

Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Miroslav Velk

## Porovnávání obrázků na základě simulace pohybu očí

Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Martin Bálek

Studijní program: Informatika

Studijní obor: Programování

Praha 2011

Moje poďakovanie patrí predovšetkým vedúcemu práce za rady a hlavne za trpezlivosť, ale tiež kamarátom za ich podporu a pomoc pri dokončovaní práce.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne

Miroslav Velk

Název práce: Porovnávání obrázků na základě simulace pohybu očí

Autor: Miroslav Velk

Katedra (ústav): Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Martin Bálek

Abstrakt: V tejto práci si predstavíme biologicky a psychologicky inšpirovaný model ľudskej vizuálnej pozornosti a vysvetlíme si význam podobných modelov. Navrhujeme a implementujeme metódy na vyhľadávanie výrazných oblastí v obrázkoch. Uvedieme detailný postup na vytváranie tzv. saliency máp, ktoré budú obsahovať informácie o výraznosti každej oblasti v skúmanej scéne. Pomocou týchto máp budeme simulovať posuny vizuálnej pozornosti (resp. pohyb očí). Vytvoríme simulovanú trajektóriu reprezentujúcu záznam týchto posunov. Tie budeme ďalej upravovať a analyzovať. Špeciálne sa zameriame na porovnávanie týchto trajektórií, a to podľa viacerých kritérií. Načrtne možnosti využitia nášho modelu v praxi.

Klíčová slova: vizuálna pozornosť, saliency mapa, vizuálne hľadanie, analýza scény, porovnávanie trajektórií

Title: Image comparison using eye movement simulation

Author: Miroslav Velk

Department: Department of Applied Mathematics

Supervisor: Mgr. Martin Bálek

Abstract: In the present work we study the biologically plausible and psychologically motivated model of human visual attention and explain the importance of similar models. We propose and implement methods to find salient locations in the image. We give detailed instructions on creating saliency maps, which contain information about saliency of every location in the explored scene. Using this maps we simulate shifts of visual attention (eye movement). A simulated scanpath representing this shifts is created and then analyzed. We especially focus on comparison of different scanpaths by different features. Finally practical use of our model is outlined.

Keywords: visual attention, saliency map, visual search, scene analysis, scanpath analysis



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Biologický základ . . . . .	1
1.2	Motivácia a prezentovaný model . . . . .	2
<b>2</b>	<b>Saliencia</b>	<b>4</b>
2.1	Výpočet saliency mapy . . . . .	4
2.2	Gaussova pyramída . . . . .	7
2.3	Mapa kontrastu intenzity . . . . .	9
2.4	Mapa kontrastu farieb . . . . .	10
2.5	Mapa zmien orientácie línií . . . . .	11
2.6	Normalizácia máp . . . . .	16
2.7	Výsledná saliency mapa . . . . .	17
2.8	Zvýraznenie centra a vignetta . . . . .	18
2.9	Ďalšie metódy vytvárania saliency máp . . . . .	19
<b>3</b>	<b>Simulovaná trajektória posunov vizuálnej pozornosti</b>	<b>20</b>
3.1	Nájdenie výraznej lokácie v saliency mape . . . . .	20
3.2	Pozdržanie návratu . . . . .	20
3.3	Ďalšie metódy . . . . .	21
<b>4</b>	<b>Porovnávanie trajektórií posunov vizuálnej pozornosti</b>	<b>22</b>
4.1	Motivácia a prezentovaný model . . . . .	22
4.2	Zjednodušenie trajektórie . . . . .	23
4.3	Zarovnanie trajektórií a porovnávanie vektorov . . . . .	24
4.4	Mapy fixácií . . . . .	26
4.5	Sila výraznosti . . . . .	27
4.6	Ďalšie metódy . . . . .	28
<b>5</b>	<b>Porovnávanie obrázkov (testy)</b>	<b>29</b>
5.1	Vstupný raster a parametre programu . . . . .	29
5.2	Automatické zisťovanie podobnosti obrazu . . . . .	30
5.3	Porovnávanie vedľa seba . . . . .	32
<b>6</b>	<b>Záver</b>	<b>35</b>
	<b>Literatúra</b>	<b>37</b>

<b>Príloha</b>	<b>39</b>
Užívateľská dokumentácia . . . . .	39
Obsah DVD . . . . .	46

# Kapitola 1

## Úvod

### 1.1 Biologický základ

Najdôležitejšou funkciou selektívnej vizuálnej pozornosti je nasmerovať náš pohľad čo najrýchlejšie na objekty záujmu v našom okolí. Schopnosť zamerať sa v rýchlosti na výrazné objekty v preplnenej scéne dáva organizmom evolučnú výhodu – dokážu rýchlo detekovať potenciálnu korisť, partnera alebo predátora vo svojom vizuálnom poli. U primátov bola zistená pozoruhodná schopnosť interpretovať komplexné scény v reálnom čase, aj napriek limitovaným možnostiam nervového vybavenia vyhradeného na túto úlohu [1]. Predpokladá sa preto existencia procesov, ktoré z dostupných senzorických informácií vyberajú podmnožinu (kvôli redukcii zložitosti scény). Takto len malá časť informácií postupuje do krátkodobej pamäte. Táto prvotná analýza nám zároveň umožňuje rozložiť problém chápania scény na menšie, výpočtovo menej náročné, lokalizované problémy vizuálnej analýzy. Tá sa uskutočňuje v priestorovo vymedzenej oblasti vizuálneho poľa, v takzvanom centre vizuálnej pozornosti („focus of attention“ odhadované na  $1^\circ$  pozorovateľného uhla). Pocit, že vidíme a vnímame všetko okolo seba (vo svojom vizuálnom poli) je teda len ilúziou.

Zameriavanie pozornosti je určené dvoma mechanizmami. Subjekt selektívne nasmeruje svoju pozornosť na objekty na základe analýzy zdola nahor (výrazné oblasti v scéne, hlavne v zmysle kontrastu so svojím okolím) a zároveň zhora nadol (v závislosti na určenej úlohe, snaha splniť cieľ). Túto myšlienku pôvodne sformuloval William James, otec americkej psychológie [2, 4].

Niektoré objekty, ako napríklad červená dopravná značka v zelenom prírodnom prostredí alebo blikajúce svetlo v inak statickej scéne automaticky a podvedome priťahujú našu pozornosť, nezávisle na zadanej úlohe a vo veľkej rýchlosti.

Pri analýze zdola nahor je prvým štádiom každého modelu vizuálnej pozornosti určenie vizuálnych príznakov (vlastností, črt) obrazu. V biologickom videní tieto vizuálne príznaky zachytávajú a rozlišujú už receptorové bunky na sietnici, bočné kolienkovité telieska (LGN - primárne prenosové centrum vizuálnych informácií zo sietnice do thalamu), Colliculi superiors (centrum nepodmienených zrkovových reflexov) a nižšia zrkovová mozgová kôra. Receptorové bunky a neuróny sú v skorých štádiách aktivované jednoduchými vizuálnymi podnetmi ako kontrast

intenzity, oponentúra farieb, smer línií, zrýchlenie pohybu (ďalej len vizuálne príznaky) a to v rôznych mierkach. Vo vyšších úrovniach nervového systému neuróny reagujú na ďalšie zložitejšie vizuálne význačné črty ako hrany a rohy, geometrické tvary až po rozpoznávanie reálnych objektov [2].

Dôležité je poznamenať, že pri analýze zdola nahor je celkový kontext (charakter celej scény) dôležitejší ako lokálna sila podnetu. Nielenže sú neuróny aktivované určitým typom kontrastu na ktorý sú citlivé, ale ich konečná aktivácia je silno určená kontextom, teda oblasťou, ktorá ďaleko prekračuje ich štandardné receptívne pole. Receptívne pole neurónu alebo receptívnej bunky je oblasť zmyslového povrchu, ktorá musí byť stimulovaná, aby vyvolala odpoveď receptoru alebo neurónu.

## 1.2 Motivácia a prezentovaný model

Model, na ktorý sa zameriavam je postavený na biologicky motivovanej architektúre, ktorá sa snaží popísať ľudskú vizuálnu pozornosť a hľadanie v scéne. Prvý takýto model (popisujúci kontrolu vizuálnej pozornosti) predstavili Koch a Ullman už v roku 1985 [2]. Podstatou modelu je vytvorenie takzvanej „saliency mapy“, topografickej 2D mapy, ktorá obsahuje informácie o výraznosti každej pozície scény. Tento prístup môžeme neskôr rozšíriť simuláciou spätnej odozvy z vyšších oblasti mozgovej kôry (napr. poznatky o cieľoch ktoré hľadáme) tak, že ohodnotíme dôležitosť jednotlivých vlastností scény.

U primátov sa predpokladá, že podobná saliency mapa vzniká v oblasti zadnej parietálnej (temennej) kôry ako aj v jadrách vankúša thalamu (Thalamus je zadná časť medzimozgu, nuclei pulvinare - zadná skupina jadier, prepojujacie jadrá pre zrakové a sluchové dráhy) [1].

Samotný model saliency máp zahrňuje určitú vnútornú dynamiku, ktorá zaisťuje posuny vo vizuálnej pozornosti. Tento model je teda schopný zaistiť kompletnú simuláciu pozornosti na základe analýzy zdola nahor a nevyžaduje žiadne riadenie zhora nadol.

Podobné modely preukázali vysokú presnosť pri detekovaní výrazných objektov v prírodných scénach, ako sú napríklad dopravné značky či chodci. Pri vyhľadávaní vojenských vozidiel na fotografiách s vysokým rozlíšením z databázy Search2 (databáza poskytnutá NATO) model v rýchlosti vyhľadávania prekonáva ľudí. Predpokladá sa, že ľudia sa zamerali na konkrétnu úlohu (nájsť tank), preto sa snažili o pochopenie scény (komplexná úloha, analýza zhora nadol), zatiaľ čo model pri analýze zdola nahor začal postupne prechádzať lokácie, ktoré sa vymykali svojmu okoliu [11].

Vývoj počítačových modelov, ktoré simulujú zameriavanie pozornosti v danej scéne môžu mať potenciálne veľké využitie v návrhu umelého videnia, automatickej detekcie cieľa, navigačných pomôcok a zariadení, robotike, automatickom orezávaní obrazu, kompresii, alebo ako si ukážeme, v posudzovaní grafického dizajnu produktov a reklám.

Po vytvorení simulovanej trajektórie posunov pozornosti je potrebné tieto trajektórie porovnávať. Na porovnávanie sa používajú rôzne metriky zohľadňujúce celkový tvar trajektórie, polohu a trvanie fixácií či smer pohybu. Pri porovnávaní trajektórií je preto potrebné zvážiť, ktoré vlastnosti nás zaujímajú.

V celom procese nesmieme zabúdať na fakt, že výsledky simulácií nemusia vždy zodpovedať reálnym posunom pozornosti. Naše výsledky sa snažia zachytiť len najnižšie úrovne ľudského videnia, bez analýzy zhora nadol. Navyše bolo zistené, že posuny pozornosti, nemusia vždy zodpovedať pohybu očí [5].

# Kapitola 2

## Saliencia

### 2.1 Výpočet saliency mapy

Väčšina modelov analýzy zdola nahor (ako aj tento) vychádzajú z modelov Kocha a Ullmana a hypotézy, že ak vygenerujeme mapy určujúce výrazné oblasti pre každý vizuálny príznak, môžeme ich následne spojiť do jednej výslednej saliency mapy. Saliency mapa je skalárna, dvojdimenzionálna mapa, ktorá topograficky reprezentuje vizuálnu výraznosť každej lokácie, bez ohľadu na to, ktorý vizuálny príznak spôsobil túto výraznosť. Nezáleží teda, či je v danej výraznej oblasti červený objekt na zelenom pozadí, alebo došlo k zmene orientácie línií. Štandardne je saliency mapa reprezentovaná ako raster s bielymi pixelmi označujúcimi najvýraznejšie oblasti, tmavšie (až čierne) pixely reprezentujú menej výrazné oblasti.

Vstupom pre výpočet saliency mapy je rastrový obrázok (typicky digitálna fotografia). Tento vstupný raster je spracovávaný vo rôznych rozlíšeniach (výška a šírka rastra sa postupne zmenšuje). Vytvoríme celkom 9 pomocných rastrov, ktorých rozmery sú voči originálu v pomere 1:1 (úroveň 0) až 1:256 (úroveň 8). Každú ďalšiu úroveň vytvárame z predošlej filtráciou (gaussovo rozmazanie) a znížením rozlíšenia. Tento súbor rastrov budeme označovať ako Gaussova pyramída.

Gaussovu pyramídu použijeme pri extrakcii jednotlivých vizuálnych príznakov a vytvorení čiastkových máp pre jednotlivé vizuálne príznaky. V našom prípade nás zaujíma kontrast intenzity, kontrast farieb a zmena orientácie línií. Gaussova pyramída nám umožní pre každý bod vypočítať kontrast s jeho okolím, a to aj v závislosti na celkovom kontexte (scéne). Dokážeme tak simulovať vlastnosti receptívneho poľa, ako sú popísané v úvode.

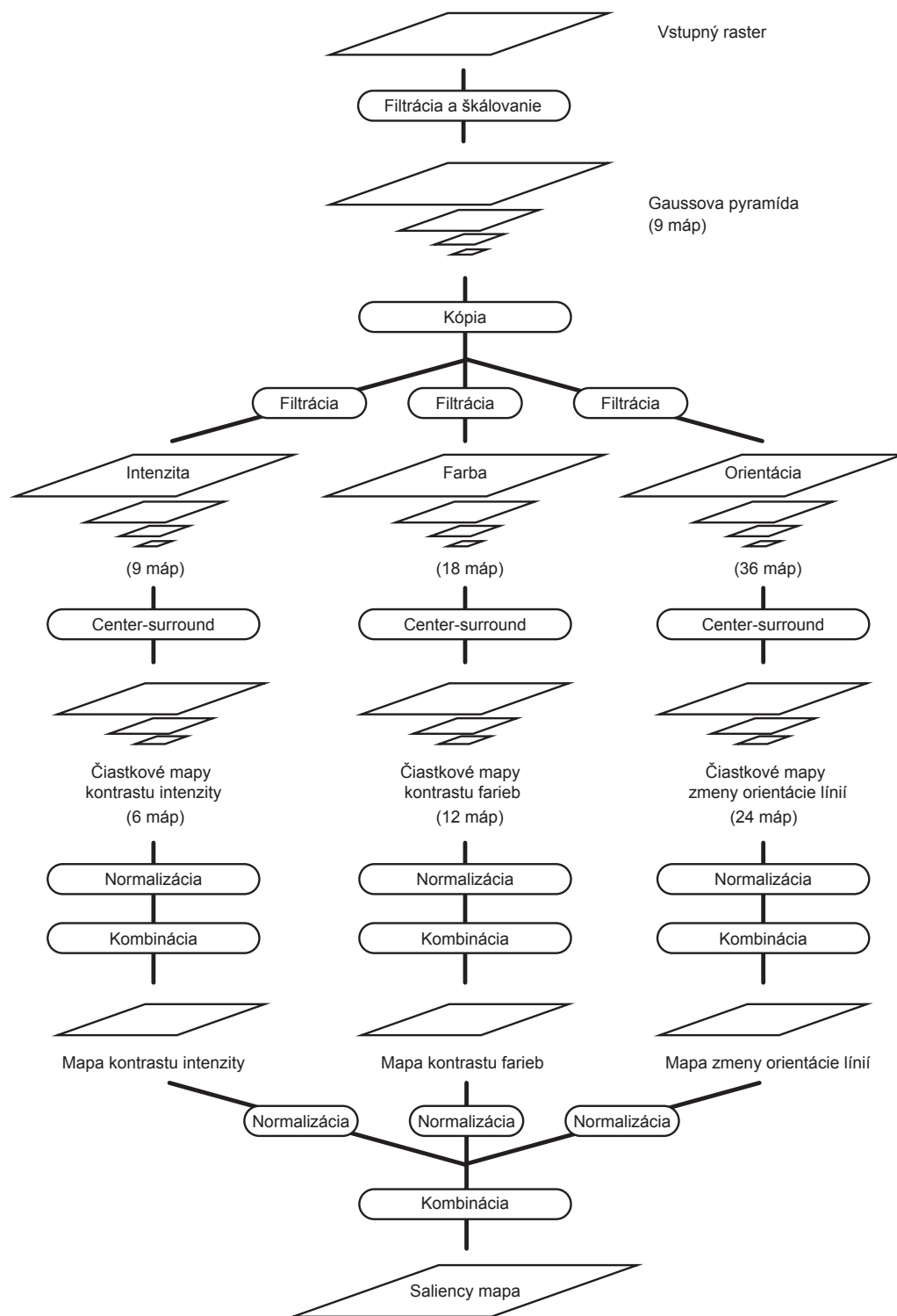
Pre extrakciu vizuálnych príznakov je potrebné na všetky úrovne Gaussovej pyramídy aplikovať príslušné filtre. Pre výpočet mapy kontrastu intenzity extrahujeme v každej úrovni pyramídy z farebného rastra intenzitu. Pre výpočet mapy kontrastu farieb extrahujeme červeno-zelený a modro-žltý kanál (farebné opponentúry). Mapu zaznamenávajúcu zmenu orientácie línií získame tak, že z každej úrovni Gaussovej pyramídy extrahujeme intenzitu (ako pri vytváraní mapy intenzity) a následne každú úroveň konvolujeme s príslušným Gabor filtrom pre danú orientáciu línií.

Po aplikácii filtrov vypočítame kontrast jednotlivých vizuálnych príznakov medzi stredom a okolím (v každom bode) ako rozdiel medzi jemnou (vysoké rozlíšenie) a hrubšou (nižšie rozlíšenie) úrovňou Gaussovej pyramídy (tzv. „center-surround“ operácia). Centrom je pixel na úrovni  $c \in \{2, 3, 4\}$  a okolie predstavujú pixely na úrovni  $s = c + \delta$ , kde  $\delta \in \{3, 4\}$ . Rozdiel medzi rôznymi úrovňami označíme ako  $\ominus$  a získame ho interpoláciou na jemnejšiu úroveň a odčítaním bod po bode (v absolútnej hodnote). Zvolením rôznych hodnôt, nie len pre  $c$ , ale aj  $\delta$ , berieme do úvahy rôzne veľkosti okolia, v ktorom hľadáme kontrast [1].

Získané čiastkové mapy kontrastov musíme následne skombinovať do jednej mapy pre každý vizuálny príznak a napokon do výslednej saliency mapy. Tieto mapy však reprezentujú vzájomne nekompatibilné vlastnosti, majú rôzny dynamický rozsah a mechanizmus extrakcie. Výrazný objekt sa môže silno prejavíť len na niektorých mapách, na iných môže byť maskovaný veľkým množstvom šumu. Objekt, ktorý je výrazný na malom počte máp môže byť tiež maskovaný menej výraznými objektmi prítomnými na viacerých mapách. Preto je pred kombináciou máp potrebné vykonať normalizáciu. Mapy je možné skombinovať až po normalizácii.

Pre zvýšenie výkonu aplikácie rozdelíme nezávislé výpočty do viacerých vlákien.

Saliency mapu môžeme okrem už popísaného postupu vytvárať aj ďalšími metódami. Náš model môžeme modifikovať použitím pôvodného postupu na výpočet mapy kontrastu farieb, ako ho uvádza pôvodný Ittyho model (využívame novší postup, ktorý dáva lepšie výsledky). Ďalšou možnosťou je využitie Gabor filtrov aj pri spracovávaní farebných kanálov. Ostatné kroky zostávajú nezmenené. Všetky spomínané postupy využívajú pri spracovávaní obrazu filtrovanie. Príkladom úplne odlišného prístupu je metóda využívajúca grafové algoritmy. Tieto metódy sú viac popísané v kapitole 2.9.



Obr. 2.1: Proces vytvárania saliency mapy.



## 2.2 Gaussova pyramída

Pre každý bod potrebujeme vypočítať kontrast s jeho okolím a to aj v závislosti na celkovom kontexte (scéne). Práve na to nám slúži Gaussova pyramída. Zo vstupného rastra sa snažíme vytvoriť 9 pomocných rastrov s rozmermi v mierke 1:1 (originál, úroveň 0), až 1:256 (najmenší rozmer, úroveň 8). Pomocou tejto pyramídy dokážeme neskôr simulovať vlastnosti receptívneho poľa, ako sú popísané v úvode [1]. Cieľom je v každej ďalšej úrovni zmenšiť rozmery rastra na polovicu a raster filtrovať (mierne rozmazať).

Pre vytvorenie pyramídy je potrebné dodržať určité podmienky a postup vytvárania. Na vstupe dostávame raster obsahujúci  $C$  stĺpcov a  $R$  riadkov. Originálny raster umiestnime v pyramíde na úroveň 0 (raster  $g_0$ ). Úroveň 1 potom obsahuje raster  $g_1$ , ktorý je redukovanou verziou  $g_0$ , a to tak, že každý pixel je spočítaný ako vážený priemer hodnôt na úrovni 0 v rámci okolia  $5 \times 5$  pixelov (zodpovedajúceho pixelu v  $g_0$ ). Hodnoty v úrovni 2 (raster  $g_2$ ) sú získané rovnako z úrovne 1 (rastra  $g_1$ ) použitím rovnakého váženého priemeru. Veľkosť okolia pritom nie je kľúčová. Oblasť  $5 \times 5$  ale poskytuje postačujúce filtrovanie pri relatívne malých nárokoch na výpočtové prostriedky.

Škálovanie môžeme definovať ako funkciu

$$g_k = REDUCE(g_{k-1})$$

a to tak, že pre úrovne  $0 < l < N$  a pre pixely  $i, j$ ,  $0 \leq i < C_l$ ,  $0 \leq j < R_l$  platí

$$g_l(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{l-1}(2i + m, 2j + n).$$

$N$  predstavuje počet úrovní pyramídy, zatiaľ čo  $C_l$  a  $R_l$  sú rozmery rastra v úrovni  $l$ . Počet pixelov je zredukovaný na polovicu v každom smere (výška, šírka), teda celkovo na štvrtinu na každej ďalšej úrovni.

Rozmery pôvodného rastra sú vhodné na konštrukciu pyramídy ak existujú čísla  $M_C$ ,  $M_R$  a  $N$  také, že  $C = M_C 2^N + 1$  a  $R = M_R 2^N + 1$ . Z tohto dôvodu sú prijímané na vstupe len rastre, ktorých rozmery spĺňajú tieto podmienky (resp. rastre iných rozmerov sú najskôr škálované na vopred zadané rozmery spĺňajúce podmienky) [6].

Pre vytvorenie každej ďalšej úrovne sa používa rovnaká funkcia (filter) pre výpočet váženého priemeru. Je definovaná ako matica o rozmeroch  $5 \times 5$  a nazýva sa generujúce jadro (kernel). Pre zvýšenie výkonu volíme túto maticu separabilnú:

$$w(m, n) = \hat{w}(m) \cdot \hat{w}(n).$$

Jednodimenzionálna funkcia  $\hat{w}$  je normalizovaná

$$\sum_{m=-2}^2 \hat{w}(m) = 1$$

a symetrická

$$\hat{w}(i) = \hat{w}(-i) \text{ pre } i = 0, 1, 2.$$

Príkladom vhodných hodnôt funkcie  $\hat{w}$  je filter  $\frac{1}{16} [1, 4, 6, 4, 1]$ . Táto funkcia blízko aproximuje Gaussovu funkciu [12].

Vďaka separabilite môžeme každú ďalšiu úroveň pyramídy vytvoriť tak, že pustíme konvolúciu najprv na riadky, potom na stĺpce pôvodného rastra. Ďalšie zrýchlenie by sme mohli dosiahnuť vytvorením ekvivalentných funkcií na výpočet váženého priemeru pre každú úroveň. Všetky úrovne pyramídy tak môžeme vytvoriť priamo z originálneho rastra. Veľkosť ekvivalentnej funkcie sa v každej ďalšej úrovni zvyšuje na dvojnásobok, rovnako ako vzdialenosť medzi priemerovanými pixelmi.

Na okrajoch rastra za hodnoty chýbajúcich pixelov (pixels, ktoré by mali prispieť do priemeru, sú za hranicami rastra) použijú aktuálne pixely (bez posunu podľa kernelu).

Pri ďalšom spracovaní budeme potrebovať vyššie úrovne pyramídy (menšie rastre) škálovať na rozmery nižších úrovní (väčšie rastre). Pixely týchto interpolovaných rastrov pritom musia presne zodpovedať pixelom na nižších úrovniach. Z tohto dôvodu je potrebné dodržiavať už pri vytváraní pyramídy presný postup a rozmery rastrov.

Definujeme funkciu *EXPAND*, ktorá je inverzná k funkcii *REDUCE*. Cieľom je interpolovať raster rozmerov  $(M + 1) \times (N + 1)$  na raster s rozmermi  $(2M + 1) \times (2N + 1)$ . Funkcia *EXPAND* je aplikovaná na raster  $g_l$  našej gaussovej pyramídy a vráti raster  $g_{l,1}$  rovnakej veľkosti ako  $g_{l-1}$ .

Nech  $g_{l,n}$  je výsledok  $n$ -násobnej interpolácie  $g_l$ . Potom  $g_{l,0} = g_l$  a

$$g_{l,n} = \text{EXPAND}(g_l, n - 1).$$

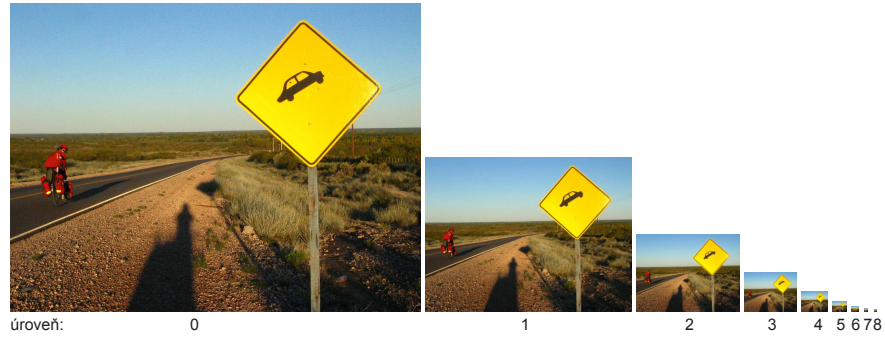
*EXPAND* definujeme ako

$$g_{l,n}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{l,n-1}\left(\frac{i-m}{2}, \frac{j-n}{2}\right),$$

kde  $0 < l \leq N, 0 \leq n$  a pre pixely  $i, j$  platí  $0 \leq i < C_{l-n}, 0 \leq j < R_{l-n}$ .

Do sumy prispievajú len prvky, u ktorých  $\frac{i-m}{2}$  a  $\frac{j-n}{2}$  sú celé čísla [6].

Na okrajoch rastra za hodnoty chýbajúcich pixelov znova použijeme aktuálne pixely (bez posunu podľa kernelu).



Obr. 2.2: Ukážka Gaussovej pyramídy o deviatich stupňoch. Rozmery rastra sa v každom kroku znižujú na polovicu.

## 2.3 Mapa kontrastu intenzity

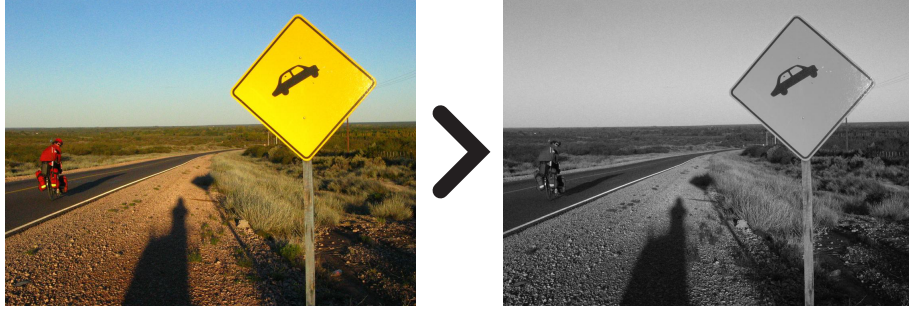
Označme červený, zelený a modrý kanál vstupného rastra ako  $r, g$  a  $b$ . Intenzitu zo vstupného rastra následne extrahujeme ako  $I = \frac{(r+g+b)}{3}$  pre každý pixel.  $I$  nazveme raster intenzity.

Na začiatok spravíme kópiu Gaussovej pyramídy. Z každej úrovne pyramídy extrahujeme intenzitu podľa predošlého vzťahu. Vytvoríme tak pyramídu  $I(\sigma)$ , kde  $\sigma \in \langle 0, 8 \rangle$  je úroveň pyramídy.

Následne na pyramídu aplikujeme už definovanú operáciu center-surround (operácia  $\ominus$ ). Tým získame, ako bolo spomínané, prehľad o kontraste intenzity každého bodu s jeho blízkym aj ďalekým okolím. U cicavcov je tento kontrast detekovaný neurónmi citlivými buď na tmavé centrum v jasnom okolí, alebo naopak jasné centrum v tmavom okolí [1, 4]. Oba typy citlivosti môžeme súčasne simulovať výpočtom šiestich máp  $I(c, s)$ , kde  $c \in \{2, 3, 4\}$  a  $s = c + \delta$ ,  $\delta \in \{3, 4\}$  takto:

$$I(c, s) = |I(c) \ominus I(s)|.$$

Pri operácii  $\ominus$  pritom používame interpoláciu zavedenú pri presnej definícii Gaussovej pyramídy.



Obr. 2.3: Príklad extrakcie intenzity zo vstupného rastra.

## 2.4 Mapa kontrastu farieb

Spravíme kópiu Gaussovej pyramídy. Z kanálov  $r$ ,  $g$ , a  $b$  extrahujeme červeno-zelené ( $RG$ ) a modro-žlté ( $BY$ ) mapy v každej úrovni pyramídy (podobne ako u máp kontrastu intenzity). Práve tieto oponentúry farieb sú totiž vnímané ako najkontrastnejšie (neuróny sú najviac citlivé na červené centrum v zelenom okolí a naopak, podobne je to s oponentúrou modrá a žltá). Na extrakciu použijeme metódu, ktorú prezentoval Walther a Koch. Táto metóda dáva lepšie výsledky ako tá, ktorú používa pôvodný Ittiho model [4]. Mapy  $RG$  a  $BY$  sú definované ako:

$$RG = \frac{r - g}{\max(r, g, b)},$$

$$BY = \frac{b - \min(r, g)}{\max(r, g, b)}.$$

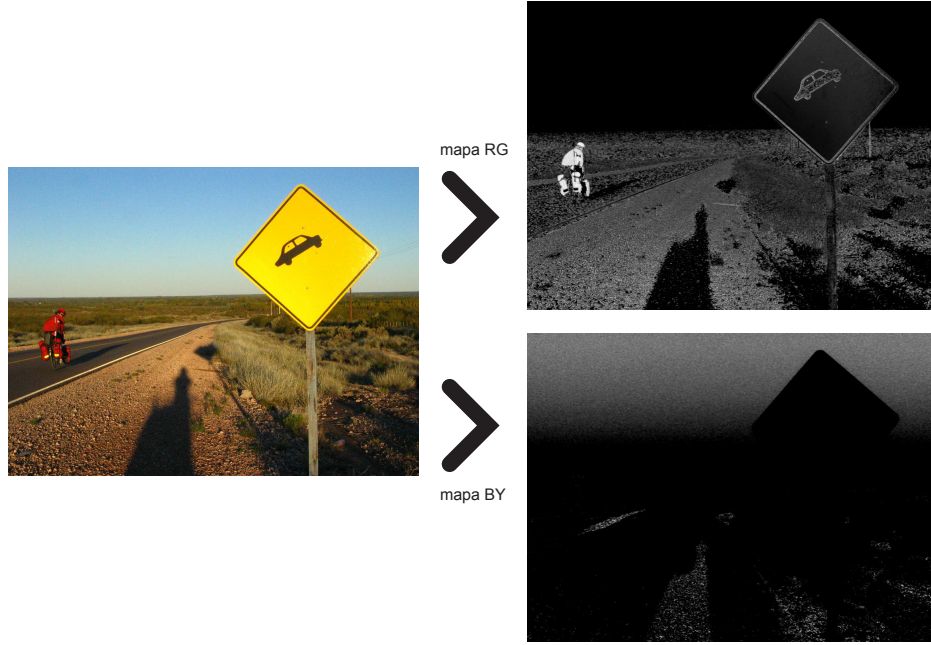
Zmeny farebných odtieňov pri veľmi nízkom osvetlení nie sú ľuďmi veľmi vnímané, a preto tieto oblasti z hľadiska farebných kontrastov nie sú výrazné. Aby sme predišli nestabilite odtieňov pri slabej intenzite (tmavé oblasti), miesta s nízkou intenzitou (kde  $\max(r, g, b) < 0,1$ ) nastavíme na 0 (čierna).

Na pyramídy  $RG$ ,  $BY$  aplikujeme operáciu center-surround rovnako ako pri vytváraní máp kontrastu intenzity:

$$RG(c, s) = |RG(c) \ominus RG(s)|,$$

$$BY(c, s) = |BY(c) \ominus BY(s)|,$$

kde  $c \in \{2, 3, 4\}$  a  $s = c + \delta$ ,  $\delta \in \{3, 4\}$ .



Obr. 2.4: Príklad extrakcie máp  $RG$ ,  $BY$  zo vstupného rastra.

## 2.5 Mapa zmien orientácie línií

Informácie o lokálnej orientácii línií získame z mapy intenzity  $I$  (ktorú dostaneme extrahovaním intenzity z pôvodného rastra). Z tejto mapy vytvoríme tzv. orientované Gabor pyramídy  $O(\sigma, \theta)$ , kde  $\sigma \in \langle 0, 8 \rangle$  sú úrovne pyramídy (podobne ako v predošlých prípadoch) a  $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$  sú uhly línií, ktoré chceme zachytiť. Vychádzame z predpokladu, že línie (alebo napríklad aj rozhrania plôch) v týchto uhloch najviac priťahujú pozornosť [1]. Vytvoríme teda 4 pyramídy o deviatich úrovniach. Na každú úroveň každej pyramídy aplikujeme 2D Gabor filter zodpovedajúcej orientácie.

Gabor filter definujeme ako funkciu 2 premenných nasledovne (zaujíma nás len reálna zložka):

$$g_{\lambda, \theta, \varphi, \sigma, \gamma}(x, y) = \exp\left(-\frac{\dot{x}^2 + \gamma^2 \dot{y}^2}{2\sigma^2}\right) \cos\left(2\pi \frac{\dot{x}}{\lambda} + \varphi\right),$$

kde

$$\dot{x} = x \cos \theta + y \sin \theta,$$

$$\dot{y} = -x \sin \theta + y \cos \theta.$$

$\lambda$  predstavuje vlnovú dĺžku. Jej hodnota je špecifikovaná v pixeloch. Validné

hodnoty sú reálne čísla väčšie nanajvýš rovné 2. Aby sa predošlo neželaným efektom na okrajoch obrázka, vlnová dĺžka by mala byť menšia ako  $1/5$  vstupného rastra. Tento parameter je možné meniť v závislosti na veľkosti vstupného rastra.



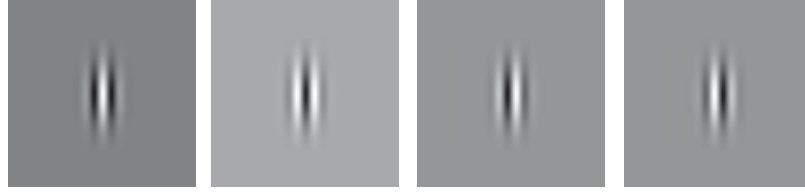
Obr. 2.5: Vizualizácia kernelu Gabor filtra s vlnovou dĺžkou 5, 10 a 15 (zľava doprava; orientácia 0; fázový posun 0; pomer strán 0,5; šírka pásma 1; obrázok prevzatý z [8]).

$\theta$  predstavuje orientáciu filtra. Hodnota sa udáva v stupňoch. Validné hodnoty sú reálne čísla z intervalu  $\langle 0, 360 \rangle$ . Tento parameter nie je možné meniť. Pri výpočte mapy orientácie línií je potrebné filtrovať pri  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  a  $135^\circ$ .



Obr. 2.6: Vizualizácia kernelu Gabor filtra s orientáciou  $0^\circ$ ,  $45^\circ$  a  $90^\circ$  (zľava doprava; vlnová dĺžka 10; fázový posun 0; pomer strán 0,5; šírka pásma 1; obrázok prevzatý z [8]).

$\varphi$  predstavuje fázový posun. Hodnota sa udáva v stupňoch. Validné hodnoty sú reálne čísla z intervalu  $\langle -180, 180 \rangle$ . Hodnoty  $0^\circ$  a  $180^\circ$  nám dávajú stredovo symetrický kernel, hodnoty  $-90^\circ$  a  $90^\circ$  kernel asymetrický. Tento parameter nie je možné meniť. Pri výpočte mapy orientácie línií je potrebné filtrovať s fázovým posunom  $0^\circ$  a  $90^\circ$ , následne výsledky skombinovať.



Obr. 2.7: Vizualizácia kernelu Gabor filtra s fázovým posunom  $0^\circ$ ,  $180^\circ$ ,  $-90^\circ$  a  $90^\circ$  (zľava doprava; vlnová dĺžka 10; orientácia 0; pomer strán 0,5; šírka pásma 1; obrázok prevzatý z [8]).

$\gamma$  predstavuje pomer strán. Určuje elipticitu výslednej Gabor funkcie.



Obr. 2.8: Vizualizácia kernelu Gabor filtra s pomerom strán 0,5 a 1 (zľava doprava; vlnová dĺžka 10; orientácia 0; fázový posun 0; šírka pásma 1; obrázok prevzatý z [8]).

$\sigma$  predstavuje smerodajnú odchýlku. Hodnotu nešpecifikujeme priamo ale pomocou kombinácie hodnôt vlnovej dĺžky  $\lambda$  a šírky pásma  $b$ . Platí:

$$b = \log_2 \frac{\frac{\sigma}{\lambda} \pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda} \pi - \sqrt{\frac{\ln 2}{2}}}, \quad \frac{\sigma}{\lambda} = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}} \cdot \frac{2^b + 1}{2^b - 1}.$$

Pokiaľ teda  $b = 1$ , tak  $\sigma \doteq 0,56\lambda$ . Čím menšia je hodnota  $b$ , tým väčšia je hodnota  $\sigma$  a tým väčší počet viditeľných pruhov v zobrazení kernelu Gabor filtra.

Pri generovaní máp zmien orientácie línií je potrebné Gabor filter aplikovať na veľké množstvo rastrov. Kernel Gabor filtra je pritom pomerne veľký. Pre urýchlenie aplikácie filtra, môžeme Gabor filter previesť do separabilnej podoby. Aby to bolo možné, zvolíme za pomer strán  $\gamma$  hodnotu 1 a uvažujeme len uhol  $\theta = 90^\circ$  [7]. Po dosadení dostaneme Gabor filter v tvare

$$g_{\lambda,\varphi,\sigma}(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma^2}\right)\right) \cos\left(2\pi\frac{x}{\lambda} + \varphi\right),$$

čo môžeme ďalej rozpísať ako

$$g_{\lambda,\varphi,\sigma}(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2}\right)\right) \cos\left(2\pi\frac{x}{\lambda} + \varphi\right) \exp\left(-\frac{1}{2}\left(\frac{y^2}{\sigma^2}\right)\right)$$

resp.

$$g_{\lambda,\varphi,\sigma}(x,y) = G_{BP}(x)G_{LP}(y),$$

kde

$$G_{BP}(x) = \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2}\right)\right) \cos\left(2\pi\frac{x}{\lambda} + \varphi\right),$$

$$G_{LP}(y) = \exp\left(-\frac{1}{2}\left(\frac{y^2}{\sigma^2}\right)\right).$$

Hodnoty  $\lambda, \varphi$  a  $\sigma$  môžeme zadať ako konštanty, získané funkcie sú teda funkciami jednej premennej ( $x$  alebo  $y$ ).

Vďaka separabilite môžeme jednotlivé úrovne Gabor pyramídy s orientáciou  $90^\circ$  vytvoriť tak, že pustíme konvolúciu najprv na riadky (s filtrom  $G_{BP}$ ), potom na stĺpce (s filtrom  $G_{LP}$ ) pôvodných rastrov intenzity. Gabor pyramídu s  $0^\circ$  orientáciou vytvoríme podobne s použitím rovnakej dvojice filtrov  $G_{BP}$ ,  $G_{LP}$  tak, že postupujeme najprv po stĺpcoch filtrom  $G_{BP}$ , následne po riadkoch filtrom  $G_{LP}$ .

Pre pyramídy so  $45^\circ$  a  $135^\circ$  orientáciou postupujeme tak, že filtre aplikujeme diagonálne. Je ale potrebné si uvedomiť, že pokiaľ filter aplikujeme po riadkoch a stĺpcoch, vzdialenosť medzi vzorkami (pixelmi) je menšia, ako pri diagonálnom prechode (uvažujeme štvorcovú mriežku/pixeli). Nech vzdialenosť stredov dvoch susedných pixelov na riadku (v stĺpci) je  $a = 1$ . Potom pixely ktoré susedia pri diagonálnom prechode majú stredy vzdialené  $a = \sqrt{2}$  [7]. Tomuto je potrebné prispôbiť aj filtre tak, že pridáme parameter  $a$  (vzdialenosť):

$$G_{BP}(x) = \exp\left(-\frac{1}{2}\left(\frac{ax^2}{\sigma^2}\right)\right) \cos\left(2\pi\frac{ax}{\lambda} + \varphi\right)$$

$$G_{LP}(y) = \exp\left(-\frac{1}{2}\left(\frac{ay^2}{\sigma^2}\right)\right)$$

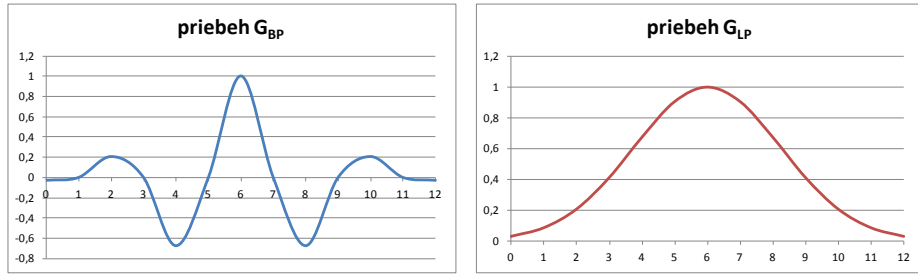


Aby sme získali lepšie výsledky, využijeme možnosti fázového posunu. Každú mapu v pyramídach filtrujeme dvakrát, a to pre  $\varphi = 0^\circ$  a  $\varphi = 90^\circ$ . Vzniknutú dvojicu pomocných máp následne normalizujeme a skombinujeme za použitia L2 normalizácie tak, že sčítame druhé mocniny hodnôt pixelov po filtrovaní a získanú hodnotu odmocníme[8].

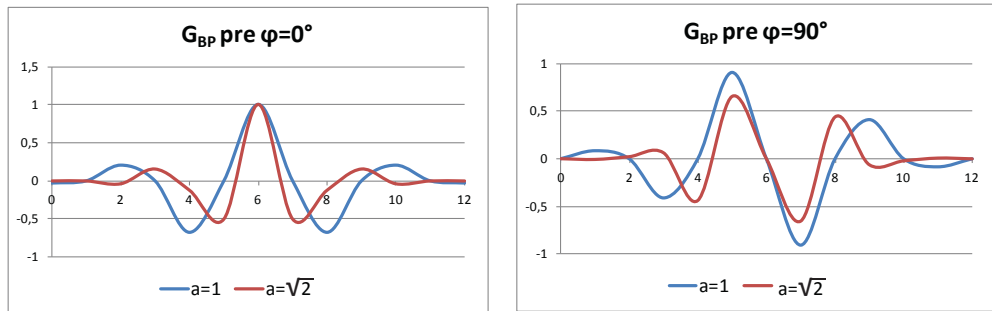
Na hotové Gabor pyramídy  $O(\sigma, \theta)$  pustíme center-surround operáciu  $\ominus$  tak, že:

$$O(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)|,$$

kde  $\sigma \in \langle 0, 8 \rangle$ ,  $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$  a  $c \in \{2, 3, 4\}$ ,  $s = c + \delta$ ,  $\delta \in \{3, 4\}$  (podobne ako v predošlých prípadoch).



Obr. 2.9: Pribeh funkcií  $G_{BP}$  a  $G_{LP}$  pri veľkosti filtra 13px ( $\lambda = 4$ ,  $\varphi = 0$ ,  $b = 1$ ):



Obr. 2.10: Pribeh funkcie  $G_{BP}$  s fázovým posunom  $0^\circ$  a  $90^\circ$  a so vzdialenosťou bodov  $a = 1$  a  $a = \sqrt{2}$ .

## 2.6 Normalizácia máp

Zavedieme normalizačnú funkciu, ktorú budeme ďalej označovať ako  $\mathcal{N}$ . Túto funkciu budeme aplikovať na mapy pred ich kombináciou (po center-surround operácii a pred skombinovaním máp pre jednotlivé príznaky).

Funkcia zabezpečí, že mapy s malým množstvom výrazných lokácií sú zvýraznené a vo výslednej kombinácii máp sa prejavajú silnejšie, zatiaľ čo mapy s veľkým množstvom porovnateľne výrazných lokácií potlačí a vo výslednej kombinácii máp sa prejavajú slabšie.

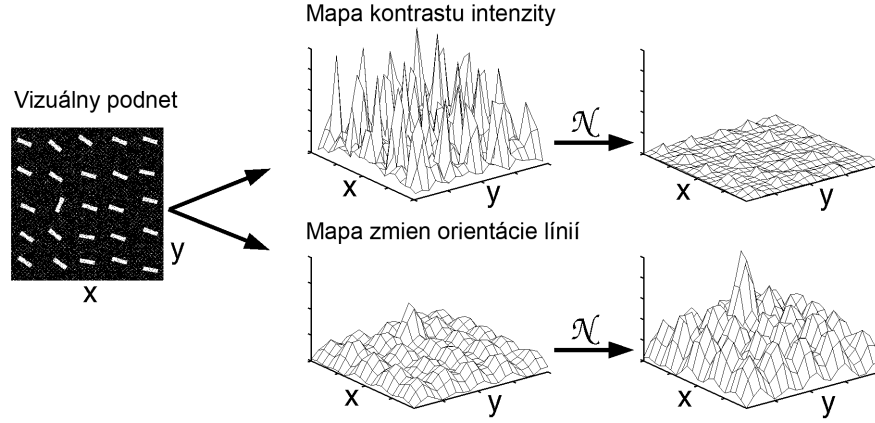
Normalizáciu vykonávame v 3 krokoch:

1. normalizujeme hodnoty v mape na fixný rozsah  $\langle 0, 1 \rangle$ ,
2. spočítame priemer  $\overline{m}$  všetkých lokálnych maxím na mape tak, že pre každý bod uvažujeme okolie štyroch (resp. ôsmich) susedných bodov; ak je hodnota skúmaného bodu maximálna voči hodnotám bodov okolia, prispieje táto hodnota do priemeru,
3. vynásobíme hodnotu každého bodu v mape hodnotou  $(1 - \overline{m})^2$ .

Ak rozdiel medzi globálnym maximom a priemerom lokálnych maxím je veľký, najvýraznejšia lokácia voči ostatným silno vyniká a mapa prispieje do kombinácie veľkou mierou. Ak je tento rozdiel malý, mapa neobsahuje žiadne unikátne výrazné lokácie a je pred kombináciou potlačená.

Pri vyhľadávaní lokálnych maxím navyše určujeme hranicu, od ktorej je možné lokáciu považovať za lokálne maximum. Pokiaľ je skúmaná hodnota pod danou hranicou, za lokálne maximum sa nepočíta. Týmto spôsobom predídeme tomu, aby len veľmi málo výrazné lokácie ovplyvnili normalizáciu mapy.

Biologickou motiváciou pre proces normalizácie je fakt, že susediace výrazné oblasti podobných vizuálnych vlastností sa navzájom utlmujú [1].



Obr. 2.11: Vizualizácia efektu normalizačnej funkcie. Zatiaľ čo mapa kontrastu intenzity obsahuje veľké množstvo podobne výrazných lokácií, mapa zmien orientácie línií obsahuje len jednu silne výraznú lokáciu (voči ostatným). Mapa kontrastu intenzity je potlačená a mapa zmien orientácie línií zvýraznená. Druhá mapa sa vo výsledku prejaví viac. Nezáleží pritom na tom, že maximum v prvej mape sú silnejšie ako maximum v druhej. Obrázok prevzatý z [1].

## 2.7 Výsledná saliency mapa

Čiastkové mapy sa po normalizácii kombinujú do máp kontrastov pre jednotlivé vizuálne príznaky. Vytvoríme teda mapu kontrastu intenzity ( $\bar{I}$ ), mapu kontrastu farieb ( $\bar{C}$ ) a mapu zmien orientácie línií ( $\bar{O}$ ). Tieto mapy získame tak, že čiastkové mapy škálujeme na rozmery štvrtej úrovne Gaussovej pyramídy a následne sčítame bod po bode (túto operáciu nazveme kombinácia máp a budeme označovať ako  $\oplus$ ).

Pre mapu kontrastu intenzity  $\bar{I}$  teda platí:

$$\bar{I} = \bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} \mathcal{N}(I(c, s))$$

Mapu kontrastu farieb  $\bar{C}$  vytvoríme podobne (navyše ale v každom kroku najprv sčítame bod po bode mapy  $RG$  a  $BY$  zodpovedajúcich rozmerov).

$$\bar{C} = \bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} [\mathcal{N}(RG(c, s) + \mathcal{N}(BY(c, s)))]$$

Pre vytvorenie mapy zmien orientácie línií  $\bar{O}$  je potrebné kombinovať oddelene mapy pre každý uhol, výsledky normalizovať a sčítat bod po bode.

$$\bar{O} = \sum_{\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}} \mathcal{N}\left(\bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} \mathcal{N}(O(c, s, \theta))\right)$$

Výslednú saliency mapu vytvoríme jednoducho, a to normalizáciou týchto troch máp a sčítaním bod po bode (následne delíme tromi, aby sme obnovili pôvodný rozsah).

$$S = \frac{1}{3} (\mathcal{N}(\bar{I}) + \mathcal{N}(\bar{C}) + \mathcal{N}(\bar{O}))$$

Motiváciou pre vytvorenie troch kanálov a ich individuálna normalizácia zodpovedá hypotéze, že každý vizuálny príznak sa snaží výraznosťou presadiť samostatne (jednotlivé vizuálne príznaky a ich vplyv prispievajú do výslednej saliency mapy nezávisle) [1].

## 2.8 Zvýraznenie centra a vignetta

Subjekt typicky zameriava svoju pozornosť na stred pozorovanej scény častejšie ako na iné lokácie. Platí teda, že čím bližšie je bod k stredu obrázka, tým vyššia pravdepodobnosť, že na neho bude zameraná pozornosť [3]. Pixely bližšie k stredu môžeme teda považovať za viac výrazné. Pre každý bod so súradnicami  $(x, y)$  vieme vypočítať jeho vzdialenosť od stredu:

$$D(x, y) = \frac{\sqrt{(x - x_0)^2 + (y - y_0)^2}}{2 \cdot \sqrt{W^2 + H^2}},$$

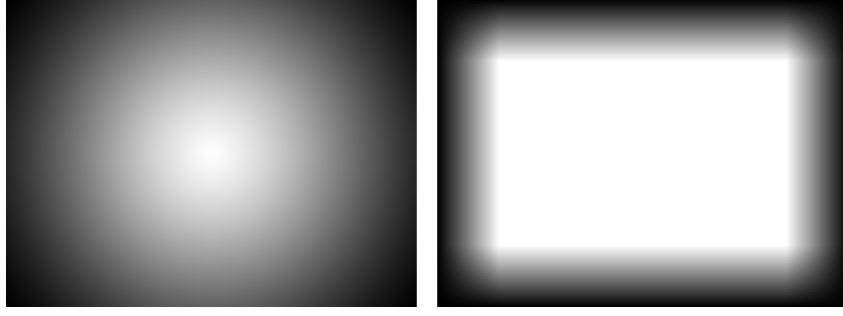
kde  $W$ ,  $H$  sú šírka a výška obrázka a  $(x_0, y_0)$  sú súradnice stredu (platí  $(x_0, y_0) = (W/2, H/2)$ ).

Výslednú saliency mapu môžeme upraviť nasledovne:

$$S_c(x, y) = (1 - D(x, y)) \cdot S(x, y)$$

Ďalšou výhodou zvýraznenia centra je, že potlačí artefakty, ktoré mohli vzniknúť pri mnohonásobnom filtrovaní a škálovaní máp (pri konvolúcií používame na okrajoch rastra vždy hodnotu aktuálneho pixelu).

Zvýraznenie centra môže ale nemusí byť žiaduce. Alternatívou na potlačenie artefaktov je preto štvorcová (obdĺžniková) vignetta, ktorá sa aplikuje na výslednú saliency mapu. Vignetta potlačí okrajové oblasti saliency mapy do zadanej vzdialenosti od okraja obrázku (v percentách z minima výšky a šírky vstupného rastra). Intenzita potlačenia smerom od okraja klesá exponenciálne.



Obr. 2.12: Spôsob potlačenia rastra zvýraznením centra (vľavo) a vignettou (vpravo).

## 2.9 Ďalšie metódy vytvárania saliency máp

Pôvodný Ittyho model využíva pri výpočte mapy kontrastu farieb mierne odlišný postup. Najprv vytvoríme 4 farebné kanály  $R$ ,  $G$ ,  $B$  a  $Y$  tak, že pre každý bod platí  $R = r - (g + b) / 2$ ,  $G = g - (r + b) / 2$ ,  $B = b - (r + g) / 2$  a  $Y = (r + g) / 2 - |r - g| / 2$  (záporné hodnoty nastavíme na nulu). Vytvoríme Gaussove pyramídy  $R(\sigma)$ ,  $G(\sigma)$ ,  $B(\sigma)$  a  $Y(\sigma)$  a center-surround operáciu aplikujeme nasledovným spôsobom:

$$RG(c, s) = |(R(c) - G(c)) \ominus (G(s) - R(s))|,$$

$$BY(c, s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|.$$

Ostatné kroky zostávajú nezmenené.

Ďalšou možnosťou je z obrázka extrahovať intenzitu ako v predošlom prípade a vytvoriť mapy  $RG$  a  $BY$ . Zo všetkých týchto máp vytvoríme Gabor pyramídy tak, že na každú úroveň ich Gaussových pyramíd pustíme Gabor filter [3]. Mapy kombinujeme ako v predošlom postupe.

Pri implementácii metódy využívajúcej grafové algoritmy vytvoríme z rastra graf, ktorého hrany sú ohodnotené na základe rozdielnosti lokácií v rastri. Tieto ohodnotenia normalizujeme a graf interpretujeme ako Markovov reťazec tak, že zavedieme ekvivalenciu medzi uzlami a stavmi a ďalej medzi ohodnoteniami a pravdepodobnosťou prechodu. Simulujeme čas, ktorý strávi náhodný pozorovateľ na jednotlivých uzloch/stavoch v nekonečnom čase. Uzly ktoré majú vysokú rozdielnosť voči okolitým uzlom budú mať vyššiu pravdepodobnosť návštevy pozorovateľom (prechody do týchto stavov sú pravdepodobnejšie ako prechody medzi podobnými susednými uzlami) [13].

## Kapitola 3

# Simulovaná trajektória posunov vizuálnej pozornosti

### 3.1 Nájdenie výraznej lokácie v saliency mape

Pomocou vytvorenej saliency mapy sa snažíme simulovať, kam pozorovateľ zameriava svoju pozornosť (kam nasmeruje pohľad). Potrebujeme odhadnúť lokáciu, ktorú si pozorovateľ všimne ako prvú, ale aj ďalšie lokácie, kam bude pozornosť presunutá. Tieto posuny počas simulácie zaznamenávame (pamätáme si polohu navštívených lokácií v poradí, v akom boli navštívené) a vytvoríme trajektóriu posunov vizuálnej pozornosti.

Nájdenie najvýraznejšej lokácie v scéne je pomocou saliency mapy jednoduché. Stačí nájsť najsvetlejší bod saliency mapy. Poloha tohto bodu v saliency mape zodpovedá polohe najvýraznejšej lokácie v pôvodnej scéne. Pokiaľ v mape nie je najsvetlejší bod určený jednoznačne (našli sme viac bodov s rovnakou hodnotou), skúmame hodnoty štyroch okolitých bodov (v okolí nájdených najsvetlejších bodov). Z najsvetlejších bodov vyberieme ten, ktorý má najväčší súčet hodnôt týchto okolitých bodov. Ak najsvetlejší bod spolu s okolím stále nie je určený jednoznačne, považujeme ich za ekvivalentné a vyberieme ľubovoľný z nich (v našom prípade prvý skúmaný).

Všetky ďalšie výrazné lokácie na vytvorenie trajektórie posunov identifikujeme rovnakým postupom. Pred každým ďalším vyhľadávaním je ale potrebné už navštívené lokácie utlmiť.

### 3.2 Pozdržanie návratu

Utlmovanie navštívených lokácií budeme nazývať ako mechanizmus pozdržania návratu. Pozdržanie návratu:

1. zabraňuje okamžitému návratu na už navštívené lokácie,
2. umožňuje simulovať posun pozornosti (aktuálne najvýraznejšia lokácia sa utlmí, čo dáva možnosť ďalším lokáciám stať sa tou najvýraznejšou).

Tento mechanizmus je inšpirovaný poznatkami z oboru psychofyziky (tá sa snaží fyzikálnymi zákonmi vystihnúť psychické deje). Experimentálne bolo zistené, že pozorovateľ sa do blízkosti už preskúmaných lokácií vracia s malou pravdepodobnosťou (resp. neskôr). Pravdepodobnejšie bude jeho pozornosť zameraná na nové, ešte nepreskúmané lokácie [1, 2, 11].

Súčasťou mechanizmu pozdržania návratu je krátkodobá pamäť. Tá určuje, ktoré lokácie boli už navštívené a umožňuje tak posun pozornosti na nové, zatiaľ nepreskúmané lokácie. Krátkodobá pamäť má obmedzenú kapacitu, pamätá si len obmedzený počet posledných navštívených lokácií. Po každom posune utlmíme práve lokácie uložené v krátkodobej pamäti. Sila utlmenia pritom (lineárne) klesá s každou ďalšou navštívenou lokáciou (posledná navštívená lokácia je utlmená maximálne, lokácia uložená v pamäti najdlhšie je utlmená najmenej).

Je dôležité poznamenať, že takýto mechanizmus pozdržania návratu je len hrubou aproximáciou príslušného biologického procesu u človeka. Bolo preukázané, že biologický mechanizmus pozdržania návratu je silne závislý na rozpoznávaní objektov, teda na analýze zhora nadol [2].

Posledným problémom zostáva, akú oblasť (veľkosť, tvar) saliency mapy po navštívení lokácie utlmiť. V ideálnych podmienkach by sme utlmili oblasť ktorá prislúcha objektu vyvolávajúceho výraznosť lokácie. Pri absencii analýzy zhora nadol ale volíme jednoduchší prístup a utlmujeme oblasti v tvare kruhu, ktorého intenzita od stredu k okraju klesá podľa gaussovej funkcie so stredom v aktuálne najsvetlejšom bode s priemerom ktorý zodpovedá  $\frac{1}{6}$  z minima výšky a šírky saliency mapy pre poslednú navštívenú lokáciu, resp.  $\frac{1}{3}$  z minima výšky a šírky saliency mapy pre ostatné navštívené lokácie v pamäti. Tieto hodnoty volíme znova na základe poznatkov z psychofyziky [1, 11].

### 3.3 Ďalšie metódy

Komplexnejšie modely využívajú pri výbere najvýraznejších lokácií a simulácii posunov pozornosti špeciálny typ biologicky motivovaných neurónových sietí (tzv. winner-take-all neurónové siete) založených na biologickom modeli neurónu (špeciálne tzv. integrate-and-fire neuróny).

Neurónová sieť vytvorí vrstvu nad saliency mapou tak, že neuróny dostávajú vstup zo saliency mapy. Potenciál neurónov nad výraznými lokáciami rastie rýchlejšie ako nad menej výraznými lokáciami. Každý neurón pracuje samostatne až do okamihu, kým niektorý (vítaz) nedosiahne určitú hranicu. To vyvolá posun pozornosti na lokáciu určenú neurónom (vítazom), resetovanie všetkých neurónov v sieti (globálne potlačenie) a nakoniec (lokálne) potlačenie navštívenej oblasti.

Pri použití neurónových sietí je typická pri vytváraní trajektórie simulácia času. Pozornosť sa presúva z aktuálne najvýraznejšej lokácie na ďalšiu po približne 30-70ms (simulovaný čas) a navštívená oblasť je utlmená na 500-900ms, ako bolo zistené z experimentov psychofyziky [11].

## Kapitola 4

# Porovnávanie trajektórií posunov vizuálnej pozornosti

### 4.1 Motivácia a prezentovaný model

Po vytvorení simulovanej trajektórie (alebo aj po experimentálnom zachytení pohybu očí) nás zaujíma, ako veľmi sa tieto trajektórie líšia po predložení rôznych obrázkov (je obvyklé porovnávať hlavne trajektórie získané simuláciou a experimentálne medzi sebou, to ale nie je predmetom tejto práce). Na základe tohto porovnania môžeme posúdiť, ako veľmi sú obrázky (scény) podobné z hľadiska toho, ako scénu vníma subjekt (kam zameria pozornosť, v akom poradí a pod.).

Porovnávanie trajektórií pritom nie je úplne triviálne. Pri porovnávaní môžeme zohľadňovať mnohé faktory, ako je napríklad samotná poloha navštívených lokácií, poradie v akom boli navštívené, ich vzdialenosti, u pokročilejších modelov trvanie fixácií. Nie všetky metódy porovnávania berú do úvahy všetky tieto faktory. Je dôležité si uvedomiť, či je pre nás dôležitejšia poloha navštívených lokácií, alebo aj poradie posunov. Môže nás zaujímať celkový tvar trajektórie, alebo či je možné nájsť v trajektóriách podobné posuny (kam a ako ďaleko sa pozornosť presunie z danej lokácie alebo jej okolia). Zvolíme preto metódy, ktoré nám dávajú čo najväčší prehľad o tom, ako veľmi sú trajektórie podobné z viacerých hľadísk.

Jedna z prezentovaných metód pracuje s trajektóriou ako so zoznamom matematických vektorov. Matematický vektor  $u = [x, y]$  nesie informácie o smere a dĺžke, ktoré sú pevne zviazané, je preto vhodný na reprezentáciu posunov pozornosti (uvažujeme, že posun pozornosti prebieha po priamke a dĺžka posunu je vzdialenosť lokácií). Lokácie na ktoré je zameraná pozornosť subjektu sa nachádzajú medzi týmito vektormi (ich poloha je začiatok a koniec vektoru). Reprezentácia trajektórie pomocou zoznamu vektorov teda zachováva:

1. tvar trajektórie,
2. dĺžku jednotlivých posunov,
3. smer jednotlivých posunov,
4. pozíciu navštívených lokácií.



V prípade potreby by bolo možné pracovať aj s dĺžkou trvania fixácií. Trajektórie porovnávame tak, že porovnávame zoznamy vektorov. Tieto zoznamy môžu mať rôznu dĺžku, je ich potrebné preto tzv. zarovnať, a to tak, že vytvoríme dvojice vektorov, ktoré sa budú porovnávať. Pri vytváraní dvojíc pritom dodržiavame poradie navštívených lokácií [10, 9].

Túto metódu môžeme modifikovať zmenou zarovnávaní vektorov. Dvojice vektorov vytvoríme tak, že hľadáme maximálnu podobnosť podľa zadanej metricky, bez ohľadu na poradie navštívených lokácií. Takýmto spôsobom nezachováme pôvodný tvar trajektórie, môžeme takto ale porovnávať individuálne posuny pozornosti.

Ďalšou implementovanou metódou na porovnávanie trajektórií je porovnávanie máp fixácií. Mapa fixácií je dvojrozmerné pole s rovnakými rozmermi ako je pôvodný vstupný obrázok, obsahujúca o každom bode informácie o jeho navštívení. Pri tomto spôsobe porovnávaní je ignorované poradie navštívených lokácií. Trajektórie, ktoré majú rovnaký priebeh ale opačný smer sú teda vyhodnotené ako totožné [9].

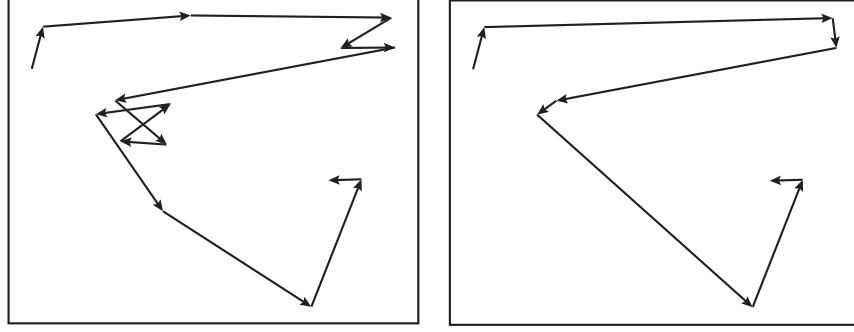
Okrem zisťovania podobnosti v trajektóriách a v polohe navštívených lokácií môžeme určiť, ktorý z porovnávaných obrázkov obsahuje výraznejšie lokácie.

## 4.2 Zjednodušenie trajektórie

Ak sa snažíme porovnávať trajektórie posunov pozornosti, nemusí byť vždy potrebné, alebo dokonca žiadané, aby trajektórie obsahovali všetky posuny. Hlavne pokiaľ ide o posuny, ktoré nemenia celkový tvar trajektórie. Trajektóriu preto môžeme často zjednodušiť. Musíme ale pritom dbať na to, aby sme zjednodušením odstránili čo najmenej dôležitých informácií. Pri prílišnom zjednodušení, trajektória stráca svoj pôvodný význam. Malé alebo žiadne zjednodušenie môže naopak znamenať problematické porovnávanie.

Často sa stáva, že pozornosť je zameraná na malú oblasť scény. Vzniká za sebou veľa krátkych posunov lísiacich sa smerom, ktoré neopúšťajú malú oblasť. Takúto trajektóriu môžeme získať ako z experimentálneho sledovania pohybu očí (po presunutí na výraznú oblasť nasleduje jej detailné skúmanie), tak aj pri simulácii. Takéto drobné posuny môžeme eliminovať bez toho, aby sme ovplyvnili celkový tvar trajektórie. Nemajú veľkú informačnú hodnotu (stačí vedieť, že subjekt nasmeroval pozornosť na danú oblasť) a ich zachovanie naopak komplikuje porovnávanie (sťažuje zarovnanie trajektórií v ďalších krokoch porovnávaní). Každú postupnosť vektorov  $\{u_1, u_2, \dots, u_m\}$ ,  $m > 1$  s dĺžkami kratšími ako  $T_{amp}$  nahradíme novým vektorom  $u' = u_1 + u_2 + \dots + u_m$ .

Ďalším spôsobom ako trajektóriu zjednodušiť, je eliminácia po sebe idúcich vektorov, ktoré majú podobný smer. Ich sčítaním nezmeníme celkový tvar trajektórie. Podobne ako v predošlom prípade, každé dva po sebe idúce vektory, ktorých smer sa líši menej ako  $T_\Phi$  radiánov, sčítame do jedného dlhšieho vektora [10].



Obr. 4.1: Príklad zjednodušenia trajektórie. Vektory s podobným smerom alebo krátkou dĺžkou (a podobnou pozíciou) sú sčítané. Vľavo pôvodná trajektória, vpravo po zjednodušení.

### 4.3 Zarovnanie trajektórií a porovnávanie vektorov

V tomto kroku máme k dispozícii dvojicu zjednodušených trajektórií. Samotné porovnávanie trajektórie spočíva v porovnávaní dvojíc zodpovedajúcich vektorov. Pre dosiahnutie lepších výsledkov je ale nutné tieto dvojice voliť vhodným spôsobom. Trajektórie sú reprezentované ako postupnosť vektorov, a teda aj ako postupnosť navštívených lokácií. Dvojice na porovnávanie vyberáme tak, aby bola zachovaná táto postupnosť (a tým aj poradie navštívených lokácií).

Zarovnanie trajektórií  $S_1 = \{u_1, u_2, \dots, u_m\}$  a  $S_2 = \{v_1, v_2, \dots, v_n\}$ , kde  $u_i$  a  $v_j$  pre  $i \leq m, j \leq n$  sú vektory, vykonáme nasledujúcim spôsobom (je dobré si všimnúť, že trajektórie nemusia mať rovnaký počet vektorov, čo je jeden z dôvodov zarovnávaní):

1. Zvolíme metriku, na základe ktorej chceme trajektórie zarovnať. Máme na to niekoľko možností. Trajektórie môžeme zarovnať na základe:

- (a) podobnosti tvaru vektorov, kde za podobnosť tvaru vektorov považujeme hodnotu (normalizujeme oddelene výškou a šírkou saliency mapy)

$$s = \frac{\left( \frac{|u(x)-v(x)|}{2H} + \frac{|u(y)-v(y)|}{2W} \right)}{2},$$

- (b) podobnosti dĺžky vektorov, kde za podobnosť dĺžky považujeme hodnotu (normalizujeme dĺžkou diagonály)

$$a = \frac{\left| \sqrt{u(x)^2 + u(y)^2} - \sqrt{v(x)^2 + v(y)^2} \right|}{\sqrt{H^2 + W^2}},$$

- (c) vzdialenosti (polohy) navštívených lokácií ( $dis(a, b)$  je vzdialenosť bo-

dov  $a, b$ , normalizujeme dĺžkou diagonály),

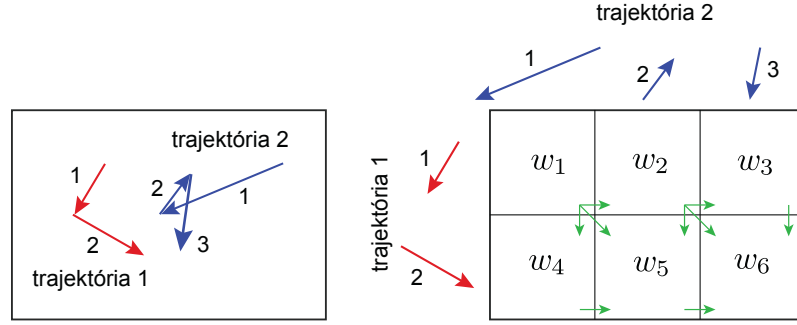
$$d = \frac{dis(a, b)}{\sqrt{H^2 + W^2}},$$

(d) podobnosti v smere vektorov ( $dir(u, v)$  je rozdiel v smere vektorov, normalizujeme uhlom  $\pi$ )

$$u = \frac{dir(u, v)}{\pi}.$$

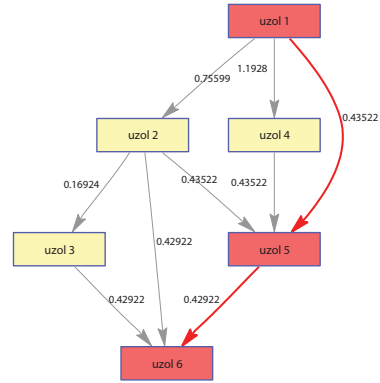
Každá z týchto metrík nám poskytne údaje o podobnosti v rozsahu  $\langle 0, 1 \rangle$ , kde 0 znamená totožnosť, 1 maximálnu rozdielnosť.

2. Podľa zvolenej metriky vypočítame, ako veľmi sú si podobné prvky  $i$  (vektory alebo lokácie) trajektórie 1 so všetkými prvkami  $j$  trajektórie 2. Nech  $\omega_{(i-1)m+j}$  je podobnosť medzi prvkami  $i$  a  $j$ . Zostrojíme maticu  $M(i, j)$  obsahujúcu všetky podobnosti (každý prvok z  $S_1$  s každým prvkom z  $S_2$ ).
3. Zostrojíme maticu susednosti  $A(k, l)$ ,  $\{k, l\} = 1, 2, \dots, mn$  z  $M(i, j)$  tak, že definujeme sadu pravidiel, podľa ktorých sú prvky matice  $M(i, j)$  prepojené (znázornené na obrázku 4.2). Každé spojenie ohodnotíme podobnosťou  $\omega_{(i-1)m+j}$ .
4. Vytvoríme graf reprezentovaný maticou susednosti (prvky matice  $M(i, j)$  sa stanú uzlami).
5. Pomocou grafu a Dijkstrovho algoritmu nájdeme najkratšiu cestu z uzla 1 (na pozícii  $[1, 1]$ ) do uzla  $mn$  (na pozícii  $[m, n]$ ). Takto zachováme poradie navštívených lokácií. Alternatívne môžeme v tomto bode zvoliť inú metódu. Nájdeme podľa danej metriky ku každému vektoru z jednej trajektórie (s väčším počtom vektorov) zodpovedajúci vektor z druhej trajektórie (s menším počtom vektorov) s maximálnou podobnosťou. Problém môže nastať, ak nie sú po tomto kroku spárované všetky vektory. Dvojice hľadáme teda aj v opačnom smere pre ešte nespárované vektory. Táto metóda nezachováva poradie, umožňuje ale porovnávať posuny pozornosti viac individuálne.
6. Trajektórie porovnávame pomocou nájdenej dvojíc vektorov. Pre každú dvojicu vektorov v zarovnaných trajektóriách určíme podobnosť podľa každej metriky. Zo všetkých hodnôt (pre danú metriku) zrátame aritmetický priemer. Výsledná hodnota určuje podobnosť trajektórií, podľa danej metriky.



Obr. 4.2: Vľavo dvojica trajektórií, vpravo matica  $M(i, j)$ . Hodnoty matice predstavujú podobnosť dvojíc vektorov. Šípky medzi hodnotami predstavujú hrany grafu ktorý sa snažíme zostrojiť. Obrázok prevzatý z [10].

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & \omega_2 & 0 & \omega_4 & \omega_5 & 0 \\ 0 & 0 & \omega_3 & 0 & \omega_5 & \omega_6 \\ 0 & 0 & 0 & 0 & 0 & \omega_6 \\ 0 & 0 & 0 & 0 & \omega_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_6 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$



Obr. 4.3: Vľavo matica susednosti  $A(k, l)$  vytvorená z matice  $M(i, j)$ . Vpravo zodpovedajúci graf s vyznačenou najkratšou cestou. Podľa tejto cesty vytvoríme zariadenie trajektórie tak, že vytvoríme dvojice vektorov 1-1, 2-2 a 2-3 (prvý vektor do dvojice vezmeme z trajektórie 1, druhý z trajektórie 2). Obrázok prevzatý z [10].

Nakoniec teda dostávame 4 hodnoty, ktoré nám popisujú podobnosť trajektórií z hľadiska ich tvaru, dĺžky posunov pozornosti, vzdialenosti navštívených lokácií a smeru prechádzania lokácií. Všetky hodnoty sú v rozsahu  $\langle 0, 1 \rangle$ , kde malé hodnoty predstavujú vysokú podobnosť, veľké hodnoty podobnosť nízku [10].

## 4.4 Mapy fixácií

Mapy fixácií nám môžu poskytnúť ďalšie informácie o podobnosti scén. Pri ich vytváraní začíname s prázdnu mapou. Každú navštívenú lokáciu zakreslíme do mapy ako kruh s pevným priemerom, ktorého intenzita od stredu k okraju klesá podľa gaussovej funkcie (gaussova funkcia so stredom v bode fixácie) [9].

Uvažujeme 2D gaussovu funkciu:

$$f(x, y) = Ae^{-\left(\frac{(x-x_0)^2}{2\sigma^2} + \frac{(y-y_0)^2}{2\sigma^2}\right)},$$

kde  $A$  je amplitúda,  $x_0, y_0$  sú súradnice stredu,  $\sigma$  nám umožňuje upravovať šírku funkcie. Na určenie parametra  $\sigma$  využijeme fakt, že  $\text{FWHM} = 2\sqrt{2\ln 2}\sigma$  (FWHM značí „full width at half maximum“, šírku intervalu v ktorom funkčná hodnota klesne z  $A$  na  $A/2$ ).

Pokiaľ umiestňujeme na navštívenú lokáciu 2D gaussovu funkciu a tá sa prekrýva s už uloženými dátami, hodnoty sčítame. Trajektórie porovnávame porovnávaním máp tak, že mapy odčítame bod po bode (v absolútnej hodnote). Dostaneme tak mapu rozdielov. Vypočítaním priemernej hodnoty zo všetkých bodov mapy rozdielov zistíme, ako veľmi sú si mapy podobné. Pred porovnávaním je potrebné mapy normalizovať (upraviť na rovnaký rozsah).

Ďalšou výhodou máp fixácií je, že ich môžeme použiť na veľmi názorný popis toho, kam by subjekt pravdepodobne zameral svoju pozornosť. Hodnoty v mape fixácií budeme považovať za hodnoty jasu v pôvodnej scéne. Z pôvodného rastra a mapy fixácií tak získame raster, na ktorom sú oblasti, na ktoré bola zameraná pozornosť svetlejšie ako ostatné, menej navštevované oblasti (nezaujímavé časti scény môžu byť zakryté úplne). Alternatívne môžeme pôvodný obrázok podľa mapy fixácií rozmazávať (čím vyššia návštevnosť, tým nižšie rozmazanie).



Obr. 4.4: Pôvodný obrázok vynásobený vygenerovanou mapou fixácií. Viditeľné sú len navštívené oblasti.

## 4.5 Sila výraznosti

Zo saliency máp môžeme ľahko zistiť, ktorý z obrázkov je výraznejší (resp. obsahuje výraznejšie lokácie), a to jednoduchým nájdením hodnoty maxima na saliency mape a vypočítaním priemeru hodnôt saliency mapy z ďalších navštívených lokácií.

Poslednou z implementovaných metód je porovnávanie „vedľa seba“, kedy spojíme 2 saliency mapy do jednej (jednu vedľa druhej) a vytvoríme simulovanú trajektóriu posunov pozornosti. Určíme, koľkokrát bol pohľad zameraný na jednotlivé polovice spojeného obrázka. Získame tak predstavu o tom, ktorý obrázok je

výraznejší. Takéto porovnávanie je ale porovnávaním dvoch nezávislých saliency máp a malo by byť brané len ako doplnková metóda.

Pokiaľ chceme zistiť, ktorý z dvoch (alebo aj viacerých) obrázkov viac priťahuje pozornosť, je vhodnejšie obrázky spojiť a analyzovať ako jednu scénu.

## 4.6 Ďalšie metódy

Alternatívne sa na porovnávanie trajektórií používajú metódy založené na práci s textovými reťazcami. Rozdeľujeme ich do dvoch kategórií, a to metódy sémantické a metódy využívajúce mriežku. Najznámejšia metóda je tzv. Levenshteinova vzdialenosť. Nad trajektóriu umiestnime mriežku, ktorej riadky a stĺpce označíme rôznymi znakmi abecedy (stĺpce napríklad označíme písmenami veľkej abecedy, riadky číslicami). Trajektóriu reprezentujeme pomocou textových reťazcov tak, že každá navštívená lokácia je kódovaná ako dvojica znakov zodpovedajúceho riadku a stĺpca. Záznam trajektórie môže vyzeráť napríklad takto: A6 - C5 - F0 - I9 - C3. Porovnávanie následne prebieha pomocnou porovnávania reťazcov. Nevýhodou tejto metódy je, že už malá zmena v trajektórii (napríklad mierne posunutie celej trajektórie) môže vyvolať veľkú (a pritom nie veľmi opodstatnenú) zmenu v reťazci. Iná zmena trajektórie nemusí vyvolať v reťazci žiadnu zmenu (napríklad mierne posunutie navštívených lokácií tak, že lokácie zachovávajú svoju polohu v mriežke) [10]. Práve voči takýmto problémom je model využívajúci vektory odolný.

Často používanou metódou je Mannanova metrika. Táto metóda zohľadňuje len polohu navštívených lokácií, nie poradie, v akom boli navštívené [10]. Metóda je značne zastaralá.

# Kapitola 5

## Porovnávanie obrázkov (testy)

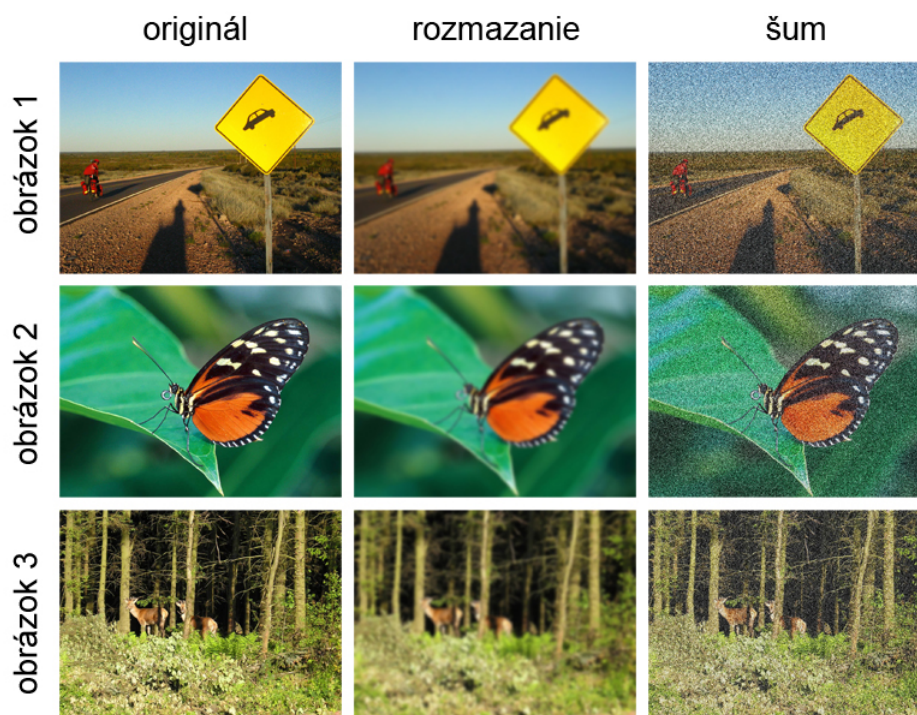
### 5.1 Vstupný raster a parametre programu

Vstupom pre porovnávanie sú obrázky (digitálne fotografie) s rozlíšením 1025x769 bodov. Porovnávame teda obrázky v relatívne vysokom rozlíšení (zvyčajne sa na vytváranie saliency máp využívajú vstupy s rozlíšením 640x480 bodov). Program prijíma aj rastre v iných rozlíšeníach, tie sú ale pred spracovaním preškálované na už spomínané rozlíšenie. Vytvárať saliency mapy je možné aj pre iné rozmery rastrov zmenou parametrov programu (to ale môže vyžadovať aj úpravu ďalších parametrov, ako sú rozmery Gabor filtra).

Pripravíme si sadu obrázkov so zaujímavými vlastnosťami, ktoré chceme skúmať. V našom prípade to budú fotografie obsahujúce zaujímavé (výrazné) objekty a k nim pripravené verzie so zníženým/zvýšeným kontrastom (oproti originálu), fotografie s pridaním šumu, gaussovo rozmazané fotografie a mierne priblížené fotografie (digitálny zoom). Tieto by mali vykazovať vysokú podobnosť s originálom. Originálne obrázky by mali medzi sebou vykazovať nižšiu podobnosť.

Ďalej si pripravíme sadu obrázkov, na ktorej chceme porovnávať výraznosť objektov. Ukážeme si, že program môže poskytnúť celkom zaujímavý pohľad na problém správneho dizajnu produktov či webových stránok.

Pri testovaní je obvyčajne postačujúce ponechať parametre programu na ich štandardných hodnotách. Pri vytváraní saliency máp vytvárame mapu kontrastu intenzity, kontrastu farieb, aj mapu zmien orientácií línií. Pri normalizácii máp považujeme bod za lokálne maximum, ak jeho intenzita presahuje aspoň hodnotu 0,05 a kontrolujeme vždy 4 okolité pixely. Na výslednú saliency mapu aplikujeme štandardne zvýraznenie stredu. Veľkosť Gabor filtra je 13px, vlnovú dĺžku nastavíme na 4, šírka pásma má hodnotu 1.



Obr. 5.1: Príklady vstupných dát z prvej sady obrázkov. Obrázky v rade by mali mať vysokú podobnosť, porovnávaním po stĺpcoch dostaneme podobnosť nižšiu.

## 5.2 Automatické zisťovanie podobnosti obrazu

Na prvej sade vstupov testujeme, ako mierna modifikácia vstupných obrázkov ovplyvní generovanie trajektórie. Na začiatok nastavíme dĺžku generovanej trajektórie na 30, pamäť pozdržania návratu na 90 krokov. Z výsledkov testovania na viacerých obrázkoch spravíme priemer.

	<i>rozmazanie</i>	<i>zvýšený kontrast</i>	<i>znížený kontrast</i>	<i>šum</i>	<i>priblíženie</i>	<i>priemer</i>
<i>tvar</i>	0,093	0,091	0,095	0,093	0,097	<b>0,094</b>
<i>pozícia</i>	0,149	0,163	0,160	0,165	0,161	<b>0,160</b>
<i>dĺžka</i>	0,117	0,115	0,117	0,127	0,123	<b>0,120</b>
<i>smer</i>	0,275	0,253	0,251	0,249	0,279	<b>0,262</b>

Tabuľka 5.1: Výsledky prvého porovnávania. Priemerná podobnosť mierne modifikovaných obrázkov. Zarovnanie podľa tvaru trajektórie.



	<i>obr. 1</i>	<i>obr. 2</i>	<i>obr. 3</i>	<i>obr. 4</i>	<i>obr. 5</i>	<i>obr. 6</i>	<i>obr. 7</i>	<i>obr. 8</i>	<i>priemer</i>
<i>tvar</i>	0,122	0,099	0,110	0,104	0,101	0,102	0,094	0,096	<b>0,104</b>
<i>pozícia</i>	0,231	0,200	0,214	0,203	0,223	0,240	0,217	0,216	<b>0,218</b>
<i>dĺžka</i>	0,160	0,127	0,145	0,119	0,119	0,135	0,116	0,137	<b>0,132</b>
<i>smer</i>	0,300	0,335	0,290	0,292	0,277	0,287	0,307	0,284	<b>0,297</b>

Tabuľka 5.2: Výsledky prvého porovnávania. Priemerná podobnosť rozdielnych obrázkov. Zarovnanie podľa tvaru trajektórie.

	<i>tvar</i>	<i>pozícia</i>	<i>dĺžka</i>	<i>smer</i>
<i>pomer</i>	1,105	1,365	1,105	1,134

Tabuľka 5.3: Pomer podobnosti medzi podobnými a rôznymi obrázkami.

Ako vidieť, podobnosť u pozmenených obrázkov je veľmi blízka podobnosti rôznych obrázkov medzi sebou. V tvare a dĺžke sú takmer rovnaké obrázky len 1,1-krát podobnejšie, v pozícii navštívených lokácií približne 1,4-krát. Skúsime zvoliť inú metódu porovnávania. Vektory zarovnáme individuálne podľa pozície (porovnáваме vektory, ktoré začínajú čo najbližšie pri sebe). Ostatné parametre zostávajú nezmenené.

	<i>rozmazanie</i>	<i>zvýšený kontrast</i>	<i>znížený kontrast</i>	<i>šum</i>	<i>priblíženie</i>	<i>priemer</i>
<i>tvar</i>	0,102	0,103	0,103	0,106	0,111	<b>0,105</b>
<i>pozícia</i>	0,022	0,024	0,019	0,026	0,028	<b>0,024</b>
<i>dĺžka</i>	0,126	0,122	0,129	0,127	0,133	<b>0,127</b>
<i>smer</i>	0,290	0,284	0,285	0,325	0,318	<b>0,301</b>

Tabuľka 5.4: Výsledky druhého porovnávania. Priemerná podobnosť mierne modifikovaných obrázkov. Zarovnanie individuálne podľa polohy vektorov.

	<i>obr. 1</i>	<i>obr. 2</i>	<i>obr. 3</i>	<i>obr. 4</i>	<i>obr. 5</i>	<i>obr. 6</i>	<i>obr. 7</i>	<i>obr. 8</i>	<i>priemer</i>
<i>tvar</i>	0,140	0,106	0,135	0,120	0,121	0,127	0,118	0,120	<b>0,123</b>
<i>pozícia</i>	0,087	0,083	0,086	0,079	0,082	0,092	0,082	0,088	<b>0,085</b>
<i>dĺžka</i>	0,185	0,130	0,149	0,136	0,139	0,153	0,145	0,130	<b>0,146</b>
<i>smer</i>	0,327	0,348	0,370	0,372	0,333	0,393	0,378	0,418	<b>0,367</b>

Tabuľka 5.5: Výsledky druhého porovnávania. Priemerná podobnosť rozdielnych obrázkov. Zarovnanie individuálne podľa polohy vektorov.

	<i>tvar</i>	<i>pozícia</i>	<i>dĺžka</i>	<i>smer</i>
<i>pomer</i>	1,177	<b>3,601</b>	1,146	1,223

Tabuľka 5.6: Pomer podobnosti medzi podobnými a rôznymi obrázkami.

Pri tomto porovnávaní sa znova ukazuje, že sada modifikovaných obrázkov vykazuje vyššiu podobnosť. V polohe fixácií je táto podobnosť dokonca 3,6-krát

vyššia, ako podobnosť rôznych obrázkov. Navyše platí, že podobnosť v tvare, dĺžke a smere je 1,18 až 1,22-krát vyššia u modifikovaných obrázkov.

Ďalej skúsime porovnávať rovnaké sady obrázkov pomocou máp fixácií. Parametre zostávajú nezmenené.

	<i>rozmazanie</i>	<i>zvýšený kontrast</i>	<i>znížený kontrast</i>	<i>šum</i>	<i>priblíženie</i>	<i>priemer</i>
<i>podobnosť</i>	0,040	0,048	0,038	0,050	0,046	<b>0,044</b>

Tabuľka 5.7: Výsledky tretieho porovnávania. Priemerná podobnosť mierne modifikovaných obrázkov. Výpočet podobnosti máp fixácií. Na zarovnaní trajektórií nezáleží.

	<i>obr. 1</i>	<i>obr. 2</i>	<i>obr. 3</i>	<i>obr. 4</i>	<i>obr. 5</i>	<i>obr. 6</i>	<i>obr. 7</i>	<i>obr. 8</i>	<i>priemer</i>
<i>podobnosť</i>	0,182	0,150	0,157	0,163	0,178	0,166	0,174	0,169	<b>0,167</b>

Tabuľka 5.8: Výsledky tretieho porovnávania. Priemerná podobnosť rozdielnych obrázkov. Výpočet podobnosti máp fixácií. Na zarovnaní trajektórií nezáleží.

	<i>podobnosť máp fixácií</i>
<i>pomer</i>	<b>3,779</b>

Tabuľka 5.9: Pomer podobnosti medzi podobnými a rôznymi obrázkami.

Podobnosť pozmenených obrázkov zrátaná z máp fixácií je asi 3,8-krát vyššia, ako podobnosť rôznych obrázkov. To približne zodpovedá porovnávaniu vektorov s individuálnym zarovnaním podľa pozície.

Ukazuje sa, že úpravy ako rozmazanie, zvýšenie či zníženie kontrastu, prídanie šumu alebo mierne priblíženie len málo ovplyvnia polohu navštívených lokácií, menia však poradie ich navštívenia. K tomuto záveru sa môžeme dostať aj vizuálnym porovnaním vygenerovaných trajektórií a máp fixácií.

Pri znížení počtu navštívených lokácií na 5 a pamäte pozdržania návratu na 15 krokov sa dostávame k podobným výsledkom, avšak rozdiely v podobnosti pre menej dát nie sú tak výrazné. K podobným výsledkom sa dostaneme aj pri vypnutí zjednodušenia trajektórií.

Kompletný záznam výsledkov porovnávania a ďalšie testy sú prílohou práce v digitálnej podobe.

## 5.3 Porovnávanie vedľa seba

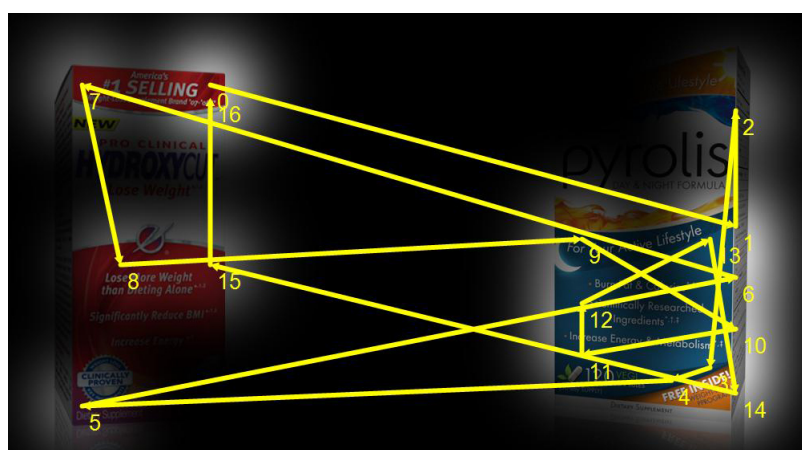
Ďalšou možnosťou porovnávania je zisťovanie, ktorý z dvojice (alebo viacerých) obrázkov je výraznejší (prípadne ktorý lepšie spĺňa naše kritéria na výraznosť). Veľmi praktickým využitím je posudzovanie dizajnu produktov, webových stránok, plagátov a podobne.

Máme pripravenú sadu fotografií znázorňujúce reálne predávané produkty a rendery produktov, ktoré sa budú snažiť tým predávaným konkurovať. Snaha je vytvoriť dizajn, ktorý bude výraznejší ako dizajn konkurencie. Ďalej máme pripravené obrázky webových stránok, na ktorých budeme sledovať, či sú požadované oblasti na stránke dostatočne výrazné a ako zmeny v dizajne, môžu ovplyvniť výraznosť.



Obr. 5.2: Ukážka časti porovnávaných produktov. Presadzovaný produkt vpravo.

Po porovnaní vedľa seba zisťujeme, že presadzovaný produkt je mierne výraznejší. Pri vytváraní simulovanej trajektórie je pozornosť zameriavaná na nový produkt takmer dvakrát častejšie, a to ako pri porovnávaní s dvoma saliency mapami, tak aj pri porovnávaní na jednom obrázku. Z mapy fixácií ďalej zisťujeme, že ľavý horný roh nového produktu je pomerne málo výrazný, čo je v rozpore s požiadavkami a mala by byť zvažovaná úprava dizajnu.



Obr. 5.3: Výsledok porovnania vedľa seba.

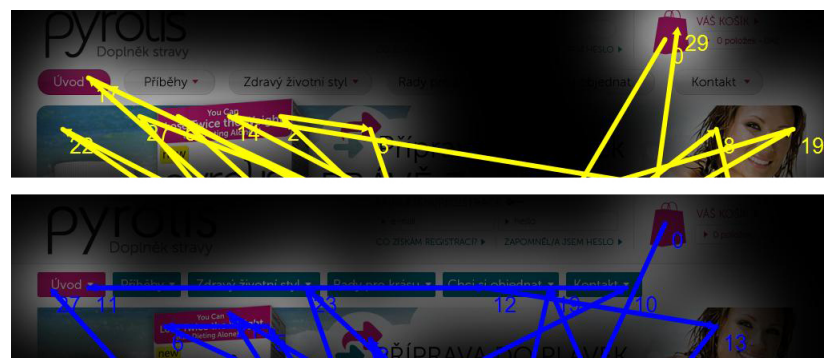
Ďalej budeme zrovnávať návrhy webových stránok.

Obe verzie boli navrhnuté so snahou naviesť užívateľovu pozornosť na banner a následne na graf v strede obrazovky. Po vygenerovaní simulovanej trajektórie



Obr. 5.4: Porovnávané návrhy webových stránok. Verzia vpravo (nový návrh) vznikla úpravou ľavej verzie (starý návrh).

nad každým návrhom zvlášť zisťujeme, že cieľ je lepšie splnený v novej verzii, kde graf dostáva dvojnásobný počet navštívení. Navyše pôvodné menu stránky nedostáva takmer žiadne návštevy. Tomu ale predídeme v novej verzii.



Obr. 5.5: Malou zmenou dizajnu dokážeme spraviť menu výraznejšie a navštevovanejšie.

# Kapitola 6

## Záver

Digitálne spracovanie obrazu je obsiahla a v súčasnosti veľmi populárna téma. Vyhľadávanie v obraze, rozpoznávanie objektov, kompresia, automatické orezávanie fotografií či porovnávanie obrázkov podľa rôznych kritérií sú úlohy, ktoré riešia každý deň milióny zariadení. Vedieť pritom, ktorá oblasť obrázka je výrazná a pre človeka preto často podstatná, môže výrazne prispieť k lepšiemu alebo efektívnejšiemu riešeniu úlohy.

Procesy, ktoré u človeka rozhodujú o tom, kam zameria svoju pozornosť sú silne podvedomé, komplikované a prebiehajú vo veľkej rýchlosti. S nárastom výkonu zariadení a s trochou dôvtipu však už vieme časť týchto procesov simulovať. Modely, ktoré sa o to snažia, sú vyvíjané už niekoľko rokov a údaje získané simuláciami sa často porovnávajú s dátami nazbieranými od užívateľov. Zdá sa pritom, že sa presnosť simulácií neustále zvyšuje.

Výsledkami ani rýchlosťou naša aplikácia zrejme neprekoná súčasne najpokročilejšie implementácie, ktoré sú schopné analyzovať video v reálnom čase a berú do úvahy pohyb objektov či kamery. Ukázali sme si však detailne základné princípy fungovania týchto modelov a poukázali na veľa zaujímavých faktov o fungovaní ľudského videnia.

Zistili sme, že pri vytváraní saliency máp máme veľa možností ako postupovať. Vybrať tú najvhodnejšiu nie je práve jednoduché a často je potrebné experimentovať. Voľba rôznych konštánt môže na prvý pohľad pripadať záhadná. Často už malá zmena konštánt alebo princípov fungovania môže viesť k úplne odlišným výsledkom.

Pri filtrovaní obrázkov sme narazili na problémy s výkonom. Väčšinu filtrov a operácií nad rastrami sme sa snažili implementovať tak, aby sme sa na dnes bežných počítačoch k výslednej saliency mape dostali v rozumnom čase (pár sekúnd).

Pri vytváraní simulovanej trajektórie vznikol problém, ako saliency mapu ďalej analyzovať. Implementovať neurónovú sieť nad saliency mapou by bolo nad rozsah tejto práce, zvolili sme preto viac priamočiary postup. Zistili sme ale, že aj náš postup dáva zaujímavé výsledky, preto sme sa s ním uspokojili.

Vygenerovali sme si simulované trajektórie posunov pozornosti a snažili sme sa ich porovnávať. Existuje veľké množstvo metód na porovnávanie a predstavili

sme aj pár vlastných. Zdá sa, že výber tej správnej je silne závislý na tom, aké parametre trajektórií a vlastnosti obrázkov chceme skúmať. Vytvoriť univerzálnu metódu alebo zvoliť automaticky tú správnu zatiaľ nie je možné. Zistili sme však, že výberom vhodnej metódy na porovnávanie môžeme odhaliť obrázky, ktoré sa líšia len malými úpravami.

Zároveň sme tak zistili, že vytváranie simulovanej trajektórie zo znehodnotených (zašumených či rozmazaných) obrázkov môže viesť k veľmi podobným výsledkom. Je dokonca možné, že trajektórie vytvorené z takýchto obrázkov, na ktorých je pre človeka ťažké rozpoznať konkrétne objekty, môžu viac zodpovedať reálnym posunom pozornosti u človeka. Odpadá totiž veľká časť analýzy zhora nadol. Človek znehodnotenú scénu menej chápe a menej sa jej snaží porozumieť.

Tiež sme prišli na to, že vypočítať podobnosť obrázkov a vrátiť výsledok ako súbor čísel nemusí byť ten najzaujímavejší postup. Ukázalo sa, že pomocou máp fixácií môžeme výsledky simulácie zobrazovať pre ľudí veľmi názornou formou. To môžeme veľmi prakticky využiť v oblasti grafického dizajnu a reklamy. Už implementované nástroje môžeme použiť ako pomôcku pri vyhodnocovaní efektívnosti s akou žiadané grafické prvky priťahujú pozornosť. Toto vyhodnocovanie samozrejme nie je možné bez účasti dizajnéra. Výsledky programu ale môžu upozorniť na potenciálne problémy, alebo potvrdiť výsledky snahy pri grafickom návrhu. Podarilo sa nám odhaliť a aj napraviť nedostatky v dizajne reálnych produktov.

Prezentovaný model je možné do budúcnosti rozšíriť. Výsledky by sme mohli značne vylepšiť pridaním segmentácie obrazu, a to bez výrazne vyšších nárokov na výpočtový výkon. Pozornosť môžeme následne presúvať po získaných segmentoch. Ako ďalšie vhodné rozšírenie sa javí rozpoznávanie geometrických tvarov a objektov, či detekcia tvárí a postáv.

# Literatúra

- [1] L. Itti, Ch. Koch, E. Niebur; a Model of Saliency-Based Visual Attention for Rapid Scene Analysis; IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 20 Issue 11, pp. 1254 - 1259, 1998.
- [2] L. Itti, Ch. Koch; Nature Reviews Neuroscience 2, pp. 194-203, 2001.
- [3] M. Wang, J. Li, T. Huang, Y. Tian, L. Duan, G. Jia; Saliency Detection Based on 2D Log-Gabor Wavelets and Center Bias; ACM Multimedia, pp. 979-982, 2010.
- [4] M. G. Quiles, D. Wang, L. Zhao, R. Romero, D. Huang; Selecting salient objects in real scenes: An oscillatory correlation model; Neural Networks 24, pp. 54-64, 2011.
- [5] J. M. Henderson, J. R. Brockmole, M. S. Castelhana, M. Mack; Visual saliency does not account for eye movements during search in real-world scenes; R. van Gompel, M. Fischer, W. Murray, & R. Hill (Eds.), Eye movements: a window on mind and brain., pp. 537-562; Oxford: Elsevier, 2007.
- [6] P. J. Burt , E. H. Adelson; The Laplacian Pyramid as a Compact Image Code; IEEE Transactions on communications, vol. COM-31, No. 4, pp. 532-540, 1983.
- [7] V. Areekul, U. Watchareeruetai, S. Tantararan; Fast Separable Gabor Filters for Fingerprint Enhancement, LNCS 3072, pp. 403-409, 2004.
- [8] N. Petkov, M.B. Wieling; Gabor filter for image processing and computer vision; on-line verzia <http://matlabserver.cs.rug.nl/edgedetectionweb/web/index.html>, 2011
- [9] D. S. Wooding; Fixation maps: quantifying eye-movement traces; ETRa '02 Proceedings of the 2002 symposium on Eye tracking research & applications, pp. 31-36, 2002.

- [10] H. Jarodzka, K. Holmqvist, M. Nyström; a Vector-based, Multidimensional Scanpath Similarity Measure; ETRa '10 Proceedings of the 2010 Symposium on Eye-Tracking, pp. 211-218, 2010.
- [11] L. Itty, Ch. Koch; Target Detection Using Saliency-based attention; RTO/SCI-12 Workshop on Search and Target Acquisition (NATO Unclassified), Volume RTO-MP-45, pp. 3.1-3.10, 1999.
- [12] K. G. Derpanis; The Gaussian Pyramid; Computer, Volume 58, Issue 4, pp. 3-4, 2005.
- [13] J. Harel, Ch. Koch, P. Perona; Graph-Based Visual Saliency; In Advances in Neural Information Processing Systems 19, pp. 545-552, 2007.



# Prílohy

## Užívateľská dokumentácia

### Požiadavky programu

Minimálne požiadavky na beh programu:

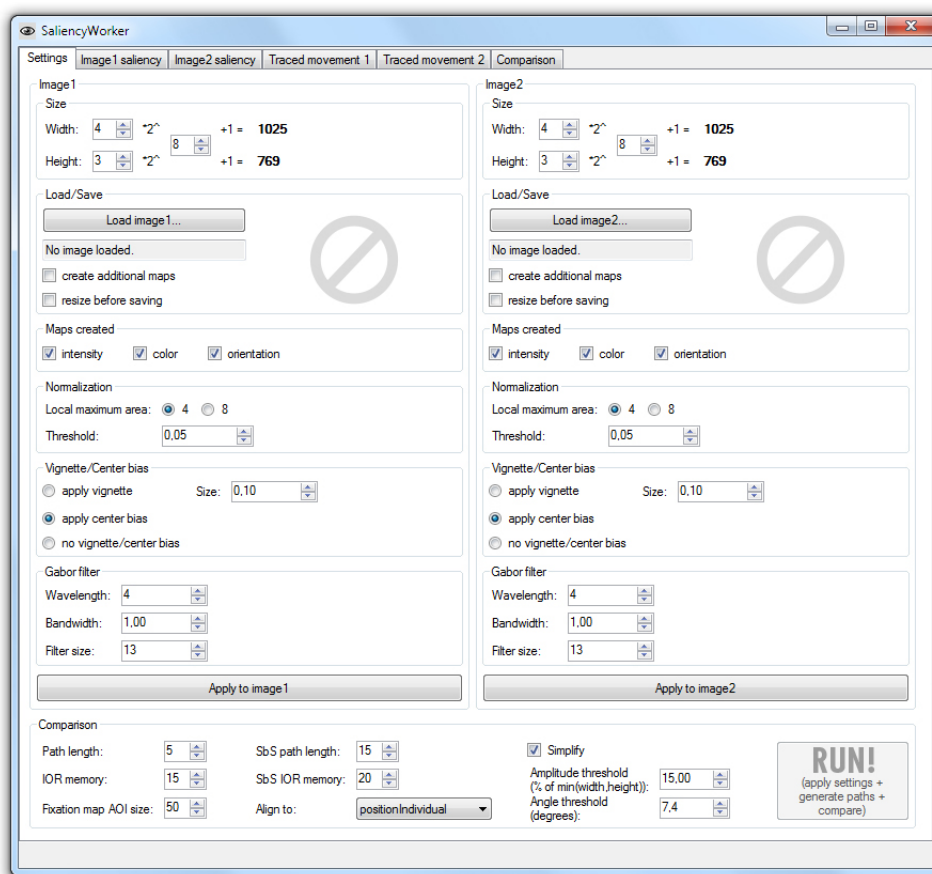
- MS Windows XP SP3, 32bit,
- MS .NET Framework 3.5,
- 2,5GB RAM,
- 2GHz jednojadrový procesor.

Odporúčané požiadavky:

- MS Windows 7, 64bit,
- MS .NET Framework 3.5,
- 4GB RAM,
- 2,5GHz viacjadrový procesor.

Program pri behu dokáže využiť viac jadier procesora pre urýchlenie výpočtu. Spotreba pamäte RAM môže pri určitej kombinácii nastavení dosiahnuť aj 1GB. Uistite sa, že máte v systéme dostatok pamäte.

## Prvé spustenie programu



Obr. 7.1: Okno programu po spustení.

Spustiteľný program nájdete na priloženom DVD

(\project\Saliency\bin\Release\Saliency.exe).

Program obsahuje grafické rozhranie na nastavovanie všetkých potrebných parametrov a vypisovanie výsledkov. Vstupom je jeden alebo dva obrázky vo formáte BMP, JPEG, GIF alebo PNG. Po spustení programu sa vám zobrazí karta „Setting“. Tá je rozdelená na 3 časti:

1. Ľavá časť označená „Image1“ obsahuje nastavenia týkajúce sa generovania saliency mapy pre vstupný obrázok 1.
2. Pravá časť označená „Image2“ obsahuje nastavenia týkajúce sa generovania saliency mapy pre vstupný obrázok 2.
3. Dolná časť označená „Comparison“ obsahuje nastavenia týkajúce sa vytvárania simulovaných trajektórií a ich porovnávania.

Ak budeme odkazovať na ovládací prvok alebo skupinu prvkov pre oba vstupné obrázky súčasne, nahradíme číslo obrázku znakom X (teda „ImageX“ znamená „Image1“ a „Image2“).

Pred začiatkom simulácie si nastavte rozmery vstupných obrázkov parametrami v skupine „Size“ (pre každý obrázok zvlášť) v nastaveniach „ImageX“. Majte pritom na pamäti, že porovnávať je možné len obrázky s rovnakým rozlíšením.

Vstupné obrázky načítate do programu kliknutím na tlačidlo „Load imageX...“ v nastaveniach „ImageX“ a zvolením príslušných súborov. Obrázky by mali mať rozmery ktoré ste si dopredu nastavili, inak prebehne automatické preškáľovanie na zadaný rozmer.

Pokiaľ si prajete len zostaviť saliency mapu pre daný obrázok, kliknite na tlačidlo „Apply to imageX“. Výsledky sa zobrazia na karte „ImageX saliency“.

Pokiaľ si prajete zostaviť saliency mapy, simulované trajektórie a v prípade načítania oboch obrázkov spustiť ich porovnávanie, kliknite na tlačidlo „RUN (apply settings + generate paths + compare)“. Výsledky sa zobrazia na rovnakých kartách ako v predošlom prípade. Navyše pribudnú výsledky generovania trajektórií na kartách „Traced movement X“ a pri načítaní dvoch obrázkov aj výsledky porovnania na karte „Comparison“.

## Pokročilé nastavenia

Grafické rozhranie vám umožňuje veľké množstvo pokročilých nastavení. Nastavenia v skupinách „ImageX“ sú totožné (umožňujú ale meniť postup generovania saliency máp každému vstupnému obrázku oddelene). Nastavenia v skupine „Comparison“ sú spoločné pre oba obrázky.

### Nastavenia v skupine „ImageX“:

- **Size:** Skupina nastavení veľkosti vstupného obrázka. Veľkosť nie je možné špecifikovať priamo, je ale vhodné priblížiť sa stranovému pomeru obrázka kombináciou čísel v týchto nastaveniach. Zmenou ľubovoľného z týchto parametrov zatvoríte aktuálne načítaný obrázok (je potrebné obrázok načítať znova).
  - *Width*: Zodpovedá  $M_C$  z kapitoly 2.2.
  - *Height*: Zodpovedá  $M_R$  z kapitoly 2.2.
  - $*2^{\wedge}$ : Zodpovedá  $N$  z kapitoly 2.2.
- **Load/Save:** Skupina nastavení k načítavaniu a zobrazovaniu obrázkov.
  - *Load imageX...*: Načítanie vstupného obrázka X.
  - *No image loaded.*: Priestor na výpis cesty k aktuálne otvorenému súboru.
  - *create additional maps*: Pri povolení aplikácia vytvorí a zobrazí vo výsledkoch dodatočné rastre, ktoré nie sú potrebné k vytvoreniu finálnej saliency mapy, môžu byť ale nápomocné pri určovaní ďalších vstupných parametrov. Povolením dodatočných rastrov sa spomaľuje výpočet.

- *resize before saving*: Škáluje pred zobrazením výsledné rastre na vyššie rozlíšenie. Zvyšuje kvalitu zobrazovania výsledkov ale spomaľuje výpočet.
- *náhľad*: Aktuálne otvorený obrázok.
- **Maps created**: Skupina určujúca, ktoré vizuálne príznaky prispievajú do saliency mapy. Pre vytvorenie saliency mapy je potrebné vytvoriť aspoň jednu mapu pre niektorý z vizuálnych príznakov.
  - *intensity*: Vytvorí mapu kontrastu intenzity.
  - *color*: Vytvorí mapu kontrastu farieb.
  - *orientation*: Vytvorí mapu zmien orientácie línií.
- **Normalization**: Skupina nastavení normalizácie pri kombinácii máp (kapitola 2.6).
  - *Local maximum area*: Skúmané okolie štyroch (resp. ôsmich) susedných bodov.
  - *Threshold*: Hranica, ktorú musí prekročiť hodnota bodu, aby mohol byť považovaný za lokálne maximum.
- **Vignette/Center bias**
  - *apply vignette*: Aplikuje na výslednú saliency mapu vignetu.
  - *apply center bias*: Aplikuje na výslednú saliency mapu zvýraznenie stredu.
  - *no vignette/center bias*: Nechá saliency mapu bez zmien.
  - *Size*: Veľkosť vignetty podľa kratšieho rozmeru vstupného obrázku.
- **Gabor filter**: Parametre Gabor filtra.
  - *Wavelength*: Vlnová dĺžka  $\lambda$ .
  - *Bandwidth*: Šírka pásma  $b$ .
  - *Filter size*: Veľkosť kernelu filtra.
- **Apply to imageX**: Vytvorí saliency mapu z načítaného obrázka podľa zvolených nastavení. Výsledky sa zobrazia na karte „ImageX saliency“.

#### Nastavenia v skupine „Comparison“:

- **Path length**: Počet navštívených lokácií pri generovaní simulovanej trajektórie nad vstupným obrázkom.
- **IOR memory**: Veľkosť pamäte pozdržania návratu (v počte krokov) pre jednotlivé obrázky.

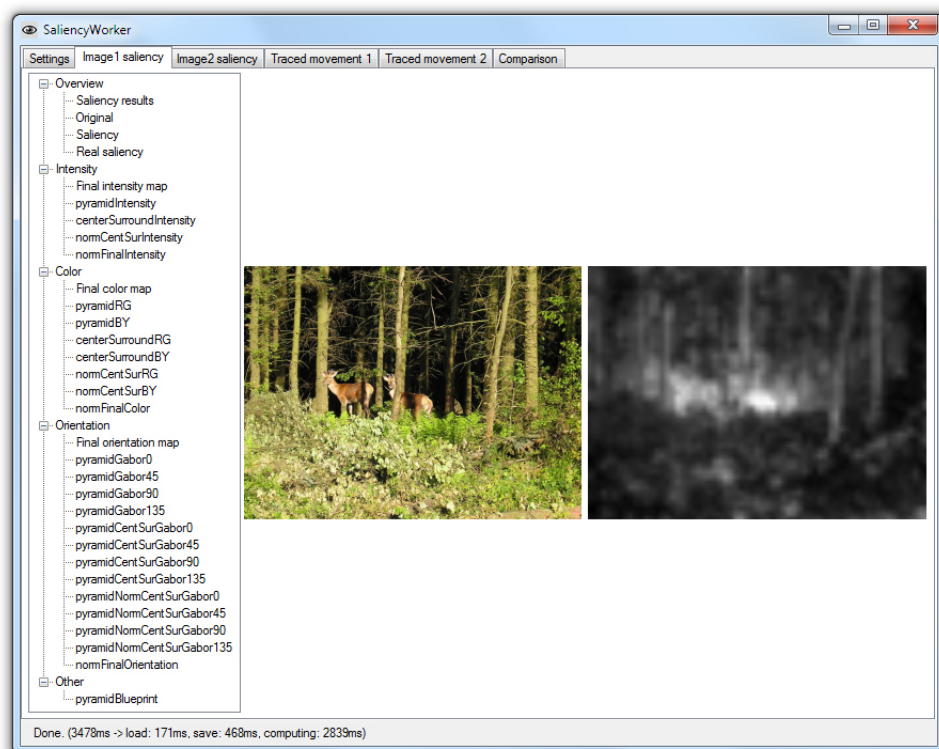
- **Fixation map AOI size:** Priemer kruhov, ktoré umiestňujeme na mapu fixácií pri jej vytváraní (kapitola 4.4).
- **SbS path length:** Počet navštívených lokácií pri generovaní simulovanej trajektórie nad oboma vstupnými obrázkami súčasne (spojené saliency mapy).
- **SaS IOR memory:** Veľkosť pamäte pozdržania návratu (v počte krokov) pre spojené obrázky.
- **Align to:** Spôsob zarovnávaní vektorov podľa
  - tvaru, pozície, dĺžky a smeru,
  - tvaru, pozície, dĺžky a smeru, bez ohľadu na poradie,
  - tvaru, pozície, dĺžky a smeru, bez ohľadu na poradie, so spárovaním každého vektoru.
- **Simplify:** Vypnutím zakážete zjednodušenie trajektórií pred ich zarovnávaním.
- **Amplitude threshold (degrees):** Hranica na spájanie krátkych vektorov pri zjednodušení trajektórií.
- **Angle threshold (% of min(width,height)):** Hranica na spájanie vektorov podobného smeru pri zjednodušení trajektórií.
- **RUN! (apply settings + generate paths + compare):** Rozhodne, pre ktorý obrázok je potrebné vytvoriť saliency mapu (pokiaľ je mapa už vytvorená a nedošlo k zmene nastavení, znova sa nevytvára), vytvorí potrebné saliency mapy pre načítané obrázky, vygeneruje simulované trajektórie pre vytvorené saliency mapy a v prípade, že sú načítané oba vstupné obrázky, spustí ich porovnanie.

## Zobrazovanie výsledkov

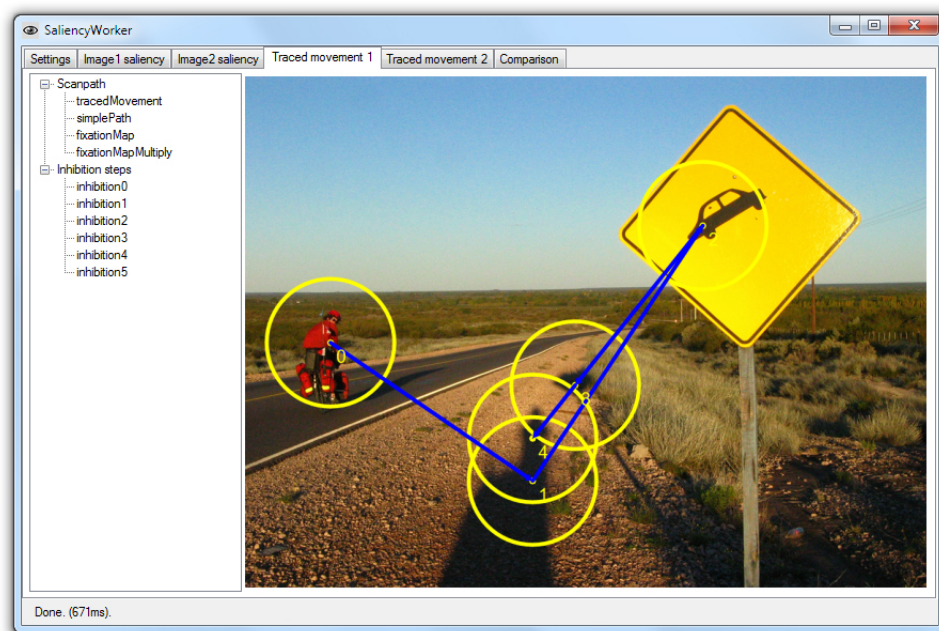
Karty „ImageX saliency“ zobrazujú výslednú saliency mapu a navyše všetky podstatné medzikroky jej výpočtu.

Veľkosť náhľadov sa automaticky prispôbuje veľkosti okna. Kliknutím na ľubovoľný obrázok ľavým tlačidlom myši obrázok zobrazíte v jeho pôvodnej veľkosti, pravým tlačidlom obrázok uložíte na disk pomocou otvoreného dialógového okna. Takéto správanie je spoločné pre väčšinu obrázkov, ktoré aplikácia zobrazuje.

Karty „Traced movement X“ zobrazujú pre konkrétny vstupný obrázok výslednú simulovanú trajektóriu, zjednodušenú trajektóriu, mapu fixácií spolu s aplikovaním na pôvodný obrázok a nakoniec saliency mapu a jej utlmenie po každom presune pozornosti.



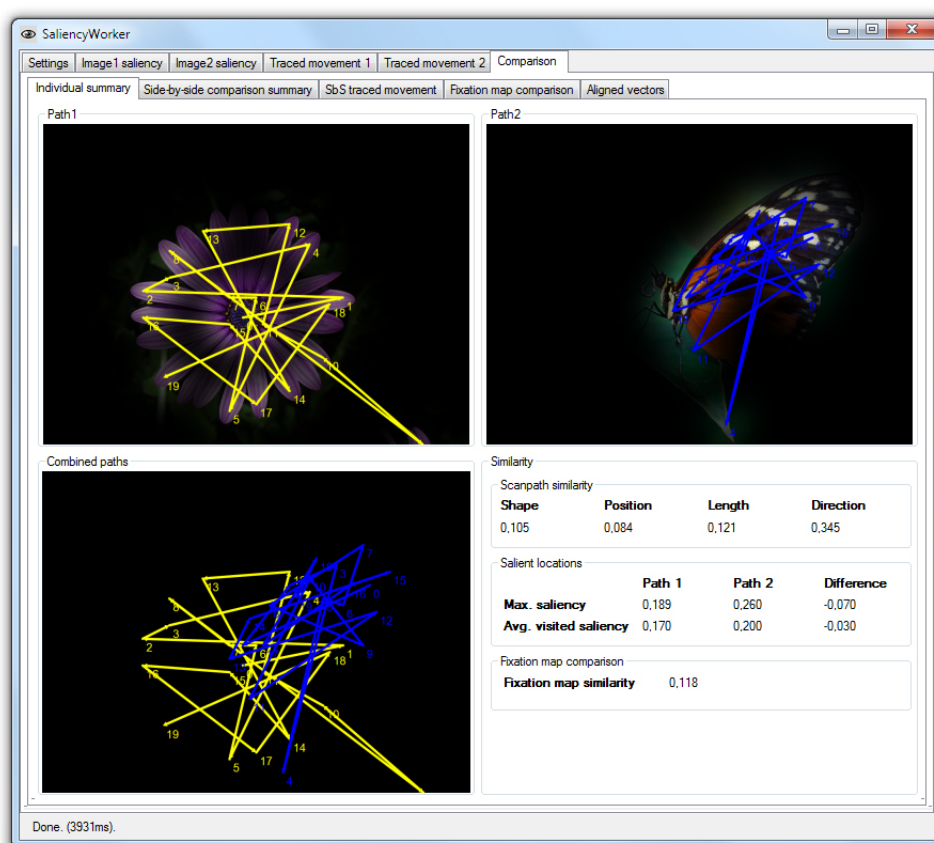
Obr. 7.2: Karta „Image1 saliency“ po vytvorení saliency mapy.



Obr. 7.3: Karta „Traced movement 1“ po vytvorení simulovanej trajektórie.

Karta „Comparison“ zobrazuje výsledky a detaily procesu porovnávání. Obsahuje dalších 5 kariet:

- **Individual summary:** Zobrazuje výsledky porovnávání vektorov podľa daných nastavení a výsledky porovnania máp fixácií.
- **Side-by-side comparison summary:** Zobrazuje výsledky porovnania vedľa seba spojením obrázkov.
- **SbS traced movement:** Zobrazuje detaily porovnávání vedľa seba spojením obrázkov.
- **Fixation map comparison:** Zobrazuje mapy fixácií nad pôvodnými obrázkami spolu s rozdielom máp fixácií.
- **Aligned vectors:** Zobrazuje spárované dvojice vektorov po zarovnaní trajektórií a ich individuálne podobnosti v tvare, pozícii, dĺžke a smere.



Obr. 7.4: Karta „Comparison“ po dokončení porovnávání. Výsledky porovnávání vektorov podľa daných nastavení a výsledky porovnania máp fixácií.

## Obsah DVD

- `\text.pdf`: Text tejto bakalárskej práce.
- `\project`: Adresár projektu pre MS Visual Studio 2008 SP1.
  - `\project\Saliency\bin\Release\Saliency.exe`: Skompilovaná aplikácia pre 32 a 64 bitové systémy.
- `\data`: Príklady vstupných obrázkov v rozlíšení, ktoré zodpovedá štandardným nastaveniam programu po spustení.
  - `\data\vectorComparison`: Testovacia sada obrázkov použitá pri zisťovaní podobnosti v kapitole 5.2.
  - `\data\productComparison`: Testovacia sada obrázkov použitá pri analýze dizajnu produktov v kapitole 5.3.
  - `\data\gaborFilterTest`: Test Gabor filtra.
- `\documentation`: Adresár obsahujúci programátorskú dokumentáciu.
  - `\documentation\HTML Help 1 file`: Obsahuje programátorskú dokumentáciu ako spustiteľný HTML Help 1 file.
  - `\documentation\webpage`: Obsahuje programátorskú dokumentáciu vo forme webovej stránky.
- `\tests`: Záznam z testovania.
- `\.NET Compact Framework 3.5 Redistributable`: Obsahuje inštalačný súbor MS .NET Framework 3.5, ktorý je potrebný na spustenie programu.